Grid Generation using the Divergence Theorem, Implicit Functions, and Constructive Solid Geometry

Eli Ateljevich [2], Phillip Colella [1], Terry J. Ligocki [*,1],
Julie Percelay [3], Peter O. Schwartz, [*,1,2]

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, California 94720 [1]

California Department of Water Resources

Abstract

We present an algorithm for calculating moments in arbitrary dimension to an arbitrary order of accuracy over regions defined by the intersection of a interface with a control volume. Such moments arise in finite volume discretizations of PDE over complex domains. The algorithm, which is adaptive and embarassingly parallel, relies on implicit function representations of surfaces, the divergence theorem, Taylor expansions, and constrained least squares. These ingredients combine in a recursion that terminates in 1D root finding and integration of monomomials along line segments.

We illustrate the algorithm using interfaces derived from image data, digital elevation maps, analytic expressions, as well as the operations of constructive solid geometry applied to of all of the above.

Key words:
PACS: 02.60.Lj, 02.70.Bf, 41.05.+e, 41.20.Cv

1 Introduction

This work describes an algorithm for grid generation within the context of embedded boundary, Cartesian grid methods. These methods use finite volume algorithms to compute numerical solutions to conservation laws, which are expressed as PDE valid over irregular domains. [**?**]. These PDE are defined in terms of a divergence of a flux, $\nabla \cdot \boldsymbol{F}$ as in the following examples:

$$\nabla \cdot \beta(\nabla u) = \rho, \;\; \boldsymbol{F}(u) = \beta(\nabla u) \qquad \text{(Elliptic PDE)}$$

$$\frac{\partial u}{\partial t} = \nabla \cdot \rho(\nabla u), \;\; \boldsymbol{F}(u) = \rho(\nabla u) \qquad \text{(Parabolic PDE)}$$

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{F}(u) = \boldsymbol{0}, \;\; \boldsymbol{F}(u) \text{ is given} \qquad \text{(Hyperbolic PDE)}$$

In this work we restrict our attention to domains, Ω, defined through an implicit function, $\phi : \mathbb{R}^D \to \mathbb{R}$. That is,

$$\Omega = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^D, \; \phi(\boldsymbol{x}) < 0\} \tag{1}$$

The Cartesian grid, embedded boundary approach discretizes the PDE on a set of control volumes, $\mathcal{V}$, formed by intersecting rectangular cells with Ω:

$$\mathcal{V} = \{[\boldsymbol{i}h, (\boldsymbol{i} + \boldsymbol{u})h] \cap \Omega : \boldsymbol{i} \in \mathbb{Z}^D\} \tag{2}$$

where $\boldsymbol{u} \in \mathbb{Z}^D$ and all its components are one and h is the size of the rectangular cells which are, in this case, cubic.

The integral of the divergence of the vector field, $\nabla \cdot \boldsymbol{F}$, over a control volume, $V \in \mathcal{V}$, can be transformed using the divergence theorem into an integral over the boundary of the volume, $A = \partial V$:

$$\int_V \nabla \cdot \boldsymbol{F} \; dV = \int_A \boldsymbol{F} \cdot \boldsymbol{n} \; dA \tag{3}$$

With these definitions, a finite volume fomulation for control volumes containing

[*] Corresponding author
Email addresses: tjligocki@lbl.gov (Terry J. Ligocki), poschwartz@lbl.gov (Peter O. Schwartz,).

[1] Supported at the Lawrence Berkeley National Laboratory by the U.S Department of Energy: Director, Office of Science, Office of Advanced Scientific Computing, Mathematical, Information, and Computing Sciences Division under Contract DE-AC03-76SF00098.
[2] Supported by the California Department of Water Resources
[3] Supported by the L'Ecole Polytechnique

embedded boundaries can be defined using the divergence theorem:

$$\int_V \nabla \cdot \boldsymbol{F} \, dV \; = \sum_{\pm\, =\, +,-} \; \sum_{d=1}^{D} \; \pm \int_{A^d_\pm} F^d_\pm \, dA \; + \; \int_{A_{EB}} \boldsymbol{F} \cdot \boldsymbol{n}_{EB} \, dA \tag{4}$$

where $A^d_\pm = \{x : x \in V, x_d = (i_d + \frac{1}{2} \pm \frac{1}{2})h\}$, $A_{EB} = V \cap \partial\Omega$, and $\boldsymbol{n}_{EB} = \nabla\phi / |\nabla\phi|$. where $\boldsymbol{n}$ is the normal to A pointing out of V. Equation 4 exhibits the divergence as a sum of fluxes integrated over faces. The numerical solution will be as accurate as the approximations to the integrals on the right hand side.

We discretize over the faces using a P^{th}-order Taylor approximations of $\boldsymbol{F}$ and $F^d_\pm$ about points, denoted $\boldsymbol{x}_{EB} \in A_{EB}$ and $\boldsymbol{x}^d_\pm \in A^d_\pm$, that approximate the centroids of the domains of integration. We canalso expand $\boldsymbol{n}_{EB}$ about some point in the cell (which we take to be the origin of our coordinates). Using these Taylor expansions, (4) can be rewritten:

$$\begin{aligned} \int_V \nabla \cdot \boldsymbol{F} \, dV \; = & \sum_{0 \le |\boldsymbol{p}| \le P} \frac{1}{\boldsymbol{p}!} \Bigg(\sum_{\pm\, =\, +,-} \; \sum_{d=1}^{D} \; \pm \, (\nabla^{\boldsymbol{p}} F^d_\pm) \int_{A^d_\pm} (\boldsymbol{x} - \boldsymbol{x}^d_\pm)^{\boldsymbol{p}} \, dA \\ & + \; \nabla^{\boldsymbol{p}} \boldsymbol{F} \cdot \sum_{0 \le |\boldsymbol{r}| \le P - |\boldsymbol{p}|} \frac{\nabla^{\boldsymbol{r}} \boldsymbol{n}_{EB}(\boldsymbol{x}_0)}{\boldsymbol{r}!} \int_{A_{EB}} (\boldsymbol{x} - \boldsymbol{x}_0)^{\boldsymbol{p}+\boldsymbol{r}} \, dA \Bigg) + O(h^{D+P}), \end{aligned} \tag{5}$$

where $\boldsymbol{x}_0$ is a point sufficiently near the interface for the Taylor remainder theorem to hold. Assuming $\boldsymbol{F}$ is discretized on the rectangular grid covered by Ω, $\nabla^{\boldsymbol{p}} \boldsymbol{F}^d_\pm$ and $\nabla^{\boldsymbol{r}} \boldsymbol{F}$ can be approximated to the appropriate order by finite differences. The remaining information required is in the form of moments, integrals of polynomials computed over V, $A^d_\pm$, and A_{EB}. This leads to an $O(h^{p-1})$ truncation error in approximating PDE.

We have shown in (5) that numerical solutions to conservation laws on irregular domains can be computed from moment calculations, finite differences of the flux functions, and quadrature. In this scheme, moment calculations play the role of grid generation in other methods for complex geometries such as methods using unstructured, mapped or multiblock grids: moment calculations summarize the information from the implicit function that is necessary for the discretization of the PDE. We used implicit functions to expand the normal in a Taylor series, but we did not require an explicit representation of the irregular boundary.

In this paper, we describe an algorithm to estimate moments. Two observations underlie our work. First, we observe that moment calculations themselves can be expressed (non-uniquely) as the divergence of a flux and therefore brought into the framework described above. Secondly, we observe that, for each degree, there is an overdetermined linear system whose unknowns represent all the moments of the given degree.

Use of this system reduces geometric and numeric complexity while increasing robustness in marginally resolved calculations. In particular, complications arising from the need to reckon the many different ways an interface can intersect a control volume are reduced to the problems of one-dimensional root finding and computing least squares solutions to overdetermined linear systems. In general, the need to work in higher dimensions or with greater accuracy implies a vast increase in difficulty. However, the algorithm we describe works in arbitrary dimension to arbitrary accuracy. The ingredients to our algorithm are implicit functions, the divergence theorem, Taylor approximations, constrained least squares, and adaptive refinement.

2 Representation of the Interface

We use implicit functions to represent the interface. Image data, digital elevation maps, and analytic expressions provide important examples of implicit functions. Constructive solid geometry applied to these examples greatly enlarges the set of possibilities, helping, for example, to represent man-made interfaces, which frequently are a combination of numerous simpler interfaces. In section 8, we describe a data structure that encapsulates a small but sufficient set of point evaluations of the implicit function and its gradients at several locations within a control volume and a data structure that holds moments over multiple dimensions. The use of templates and associative arrays facilitates some recursive aspects of our algorithm that we describe below. Moments and ratios of moments are subject to obvious constraints: volume can't be negative and centroids lie within the convex hull, for example. Specific applications may require further constraints. Section 3 descibes the formal construction of an overdetermined linear system, the ingredients of which are the divergence theorepm and a Taylor approximation to the vector normal to the interface. The solution vector estimates all the moments of a given order over a given control volume as well as moments of one higher order over the interface. The accuracy of the estimate depends on dimension, the order of the moment, and the accuracy of the Taylor approximation.

We wish to estimate moments on the discretization of an irregular domain, Ω, defined implicitly using an "implicit" function, $\phi : \mathbb{R}^D \to \mathbb{R}$, i.e.:

$$\Omega = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^D,\ \phi(\boldsymbol{x}) < 0\} \tag{6}$$

and

$$\Omega_0 = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^D,\ \phi(\boldsymbol{x}) = 0\} \tag{7}$$

Typically, Ω will be discretized on a set of control volumes, $\mathcal{V}$, formed by intersecting rectangular cells with Ω and Ω_0:

$$\mathcal{V} = \{[\boldsymbol{i}h, (\boldsymbol{i} + \boldsymbol{u})h] \cap \Omega : \boldsymbol{i} \in \mathbb{Z}^D\} \tag{8}$$

where $\boldsymbol{u} \in \mathbb{Z}^D$ and all its components are one and h is the size of the rectangular cells. Implicit function representations have several advantages:

- The implicit function, $\phi(\boldsymbol{x})$, is defined everywhere in $\mathbb{R}^D$ and thus Taylor expansions of ϕ and functions of ϕ (e.g., $\boldsymbol{n}(\boldsymbol{x}) = \nabla\phi(\boldsymbol{x})/|\nabla\phi(\boldsymbol{x})|$) can be computed anywhere they exist if ϕ is smooth enough.
- Implicit functions can be used to represent a rich set of geometric shapes through analytic expressions, interpolants of discrete, sampled data, or through constructive solid geometry.
- Implicit functions can be easily restricted to lower dimensions, which is useful for our recursion.
- Implicit functions can be extended to arbitrary dimensions allowing computations to be done in phase spaces or space-time.
- Ω can implicitly evolve in time if ϕ is allowed to change with time.

2.1 Constructive Solid Geometry

Simpler implicit functions can be composed into more complex implicit functions using constructive solid geometry, CSG. To do this, it is necessary to define the complement, intersection, and union of irregular domains, Ω_i, defined by implicit functions, ϕ_i. This is done using the following correspondences:

$$\begin{array}{lcl} \{\boldsymbol{x} : \boldsymbol{x} \notin \Omega_i\} & \Leftrightarrow & -\phi_i \\ \{\boldsymbol{x} : \boldsymbol{x} \in \cup_i \Omega i\} & \Leftrightarrow & \max\limits_i \phi_i \\ \{\boldsymbol{x} : \boldsymbol{x} \in \cap_i \Omega_i\} & \Leftrightarrow & \min\limits_i \phi_i \end{array}$$

Further, coordinate transformations, ψ, of Ω can be implemented as:

$$\Omega_\psi = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^D,\ \phi(\psi^{-1}(\boldsymbol{x})) < 0\}$$

Examples of ψ include rotations, translations, and scaling.

3 Linear System

Consider $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{p} = (p_0, p_1, \cdots, p_{D-1}) \in \mathbb{Z}^D$. Let $\boldsymbol{e}^i \in \mathbb{Z}^D$ represent the element with a one in the i^{th} position and zeros in $D-1$ remaining positions. Define $\boldsymbol{r}' = \boldsymbol{r} + \boldsymbol{e}^0$.

A basic observation that underlies our formulation is that the monomial $\boldsymbol{x}^{\boldsymbol{p}}$ can be represented (non-uniquely) as the divergence of a vector field,

$$\boldsymbol{F} = \frac{1}{p_d + 1} \boldsymbol{x}^{\boldsymbol{p}'} \boldsymbol{e}^d, \tag{9}$$

where $d \in \{0, 1, \cdots D - 1\}$ and $\boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{e}^d$

If we apply (4) to $\boldsymbol{F}$, and make a Taylor approximation of order R to the normal, we see that:

$$\begin{aligned} \int_V \boldsymbol{x}^{\boldsymbol{p}} \; dV \; - n^d(\boldsymbol{x}_0) \int_{A_{EB}} \boldsymbol{x}^{\boldsymbol{p}'} \; dA = \Bigg(& \sum_{\pm \, = \, +,-} \pm \int_{A^d_\pm} \boldsymbol{x}^{\boldsymbol{p}'} \; dA \\ & + \sum_{1 \le j \le R} \frac{\nabla^j n^d(\boldsymbol{x}_0)}{j!} \int_{A_{EB}} (\boldsymbol{x}^d - \boldsymbol{x}_0^d)^j \boldsymbol{x}^{\boldsymbol{p}'} \; dA + O(h^{p+D+R}), \Bigg) \end{aligned} \tag{10}$$

where $\boldsymbol{x}_0$ is a point sufficiently close to the interface for the Taylor remainder theorem to hold. We adopt the point of view that the two integrals on the left hand side are unknowns, which we have written as a linear combination of moments of the same order over lower dimensions and higher order moments over the interface in the current dimension. Assume for the moment that the right hand side is known. We have exhibited one equation in two unknowns. However, in dimension $D > 1$, there are two parameters, d and F, that correspond to additional equations and additional unknowns. In dimension one, the interface integral equals zero and equation 10 reduces to the fundamental theorem of calculus for monomials.

In particular, assume that we wish to calculate all moments of degree p. Let $N(p)$ denote the number of monomials of degree p in dimension D. For a given dimension, $D > 1$, there are $DN(p)$ equations of the same form a equation 10, each corresponding to a partial derivative in the x_d-direction. On the other hand, these equations use $N(p) + N(p-1)$ unknowns: $N(p)$ unknowns corresonding to the integrals over the volume and $N(p-1)$ unknowns corresponding to integrals over the interface. Thus, by calculating all the moments of degree p at once, we create an overdetermined linear system.

For example, in dimension $D = 3$ the calculation for moments of degree 2 corresponds to a matrix of size 18x9. In general, the function $N(p)$ in dimension D can be expressed as:

$$N(p) = \frac{(D-1+p)!}{(D-1)!p!} \tag{11}$$

4 Recursion

We have an exhibited an over-determined linear system, assuming the right hand side is known. We now show our assumption is legitimate, given a particular organization of the work.

For the purposes of illustration, consider the case $D = 2$. Assume p is given and R are given, and that R satisfies the equation $R = q - p - 2$, where q is the desired order of accuracy. Let G^d denote a moment of degree $p + R$.

We observe that the estimate:

$$0 = \int_{A_{EB}} (\boldsymbol{x}^d - \boldsymbol{x}^d_{_{EB}}) G^d \, dA \tag{12}$$

is sufficiently accurate. Thus we may calculate moments of degree $p+R$ using only the one-dimensional information corresponding to the first term on the right hand side. Similarly, having calculated moments over the interface of degree $p + R$ to qth-order, we may combine these with one-dimensional information to calculate all moments of degree $p + R - 1$ to the same qth-order accuracy. Continuing this process, we can recursively calculate all the needed moments over the interface, relying on the fact that a moment of one degree higher uses a Taylor approximation with one fewer term to achieve qth-order accuracy.

We have shown that, subject to our ability to solve over-determined linear systems, we can estimate moments to any degree of accuracy in two dimensions. In three dimensions, the two dimensional information provides the first term on the right hand side and a similar recursion in the degree of moments provides the second term of the right hand side.

For arbitrary dimension, D, and arbitrary degree, p, and desired order of accuracy, q, if we compute from lower dimensions to higher and from higher degree to lower, we can create a sequence of over-determined linear systems. The solution vector for a given system in our sequence represents qth-order estimates of moments of degree $p + R$ and lower over the D-dimensional volume, its boundaries, or the boundaries of its boundaries, and so on, down to one-dimensional coordinate aligned line segments.

5 Least Squares

Having created our linear system, we wish to use least squares to estimate the moments. Using least squares to calculate all the moments of a given order over a domain provides an advantage that many possible special cases and numeric complications are handled cleanly. Furthermore, the ability to impose linear constraints on the solution vector removes the need to post-process the solution to guarantee physical properties, such as the condition that moments of even degree are positive.

Indeed, when the implicit function is under-resolved or the interface lies tangent to cell faces, the algorithm can produce negative volumes or place a centroid outside of the cell. We introduce constraints as a method of enforcing a priori or application-specific information about the moments. Our goal is to ensure reasonable values without refining h. The most general bounds that we enforce are the bounds on a monomial implied by full quadrature on all the negative and positive contributing regions to the monomial in the cell V^- and V^+, which are rectangular:

$$\int_{V^-} \boldsymbol{x}^{\boldsymbol{p}-\boldsymbol{e}_d}\, dV^- <= \int_V \boldsymbol{x}^{\boldsymbol{p}-\boldsymbol{e}_d}\, dV <= \int_{V^+} \boldsymbol{x}^{\boldsymbol{p}-\boldsymbol{e}_d}\, dV^+ \tag{13}$$

When the integrand $\boldsymbol{x}^{\boldsymbol{p}-\boldsymbol{e}_d}$ is zero-degree, this constraint ensures that volume and areas fractions lie between zero and one. Constraining the higher degree monomials prevents large values of higher degree monomials from influencing the later lower-degree calculations.

We modify the bounds on the zero degree monomials to ensure the following relationship between them and all the (previously calculated) moments of degree $\boldsymbol{p} > 0$:

$$\int_V \boldsymbol{x}^0\, dV \min_V(\boldsymbol{x}^{\boldsymbol{p}}) <= \int_V \boldsymbol{x}^{\boldsymbol{p}}\, dV <= \int_V \boldsymbol{x}^0\, dV \max_V(\boldsymbol{x}^{\boldsymbol{p}-\boldsymbol{e}_d}) \tag{14}$$

This constraint serves two purposes. First, it places face centroids within their bounds. Second, it imposes a lower bound on volumes and areas that is greater than zero rather than strictly zero.

Applications may invoke new assumptions that facilitate tighter bounds. Figure *** illustrates the "no undercut bank" constraint that is useful in shallow water geometry calculations. The volume is constrained by the area of the top face times the depth of the cell. The constraint ensures that the average depth of the cell (volume over area) is within the cell. Because the constraint is a relationship between dimensions, in our algorithm it must be imposed on the higher dimension.

The set of active constraints can also be used as a criterion of fit in a local refinement scheme, alongside residuals error. We are developing a scheme in which the constraints serve both a refinement criterion and as a recourse after a maximum level of refinement is exhausted.

Our implementation of constraint least squares was carried out using QuadProg++, a freeware implementation of the Goldfarb-Idnani algorithm [GI83] for convex quadratic programming with linear equality and inequality constraints implemented for small-medium sized problems. We selected this implementation in part due to its minimal dependencies on larger libraries. The GF algorithm is an active set method, so we will be able to identify activate constraints for use as a refinement criterion. The general linear constraints have not proved necessary and we are considering the performance benefits of switching to an active set least squares solver with only bound constraints.

6 More on Least Squares

Each equation in our system is accurate to $o(D+p+R)$ and hence the computed least squares solution will have the same accuracy, provided our system is well-conditioned. Indeed, the condition number of our system is independent of h and independent of the direction of the normal(cite example matrix and calculation). Moreover, under least squares, the error for each equation tends to be roughly the same in absolute terms as well as being the same order. Unfortunately, in special cases the exact answer to some equations in the system can be approximately the same as the error. That is, the relative error can be large. For example, if the volume is small, but the implicit function is changing rapidly, the error arising from the Taylor approximation can be the same magnitude as the volume, which is not a satisfactory calculation. This is particularly likely to happen, when the point of expansion in the Taylor series is farther from the interface and particularly likely to happen when different components of the solution vector are of widely different magnitudes. For example, a first moment on the boundary can often be large compared to the zeroth moment in the volume as in figure??? Both of these quantities can are computed in a single least squares problem, which leads to an estimated solution with large relative error in the zeroth moment and small relative error in the first moment.

Refinement and the use of constraints mitigate this problem. Refinement does more work to get a better answer and constraints do a little extra work to prevent an answer from violating common sense. However, there is a third improvement, also requiring only slightly more work, that we wish to describe.

We introduce another step in the algorithm: moving the origin in a moment calculation. Using the binomial theorem we represent a moment of the form:

$$\int_V (\boldsymbol{x} - x_0)^{\boldsymbol{p}} \, dV \tag{15}$$

as a sum of moments of the form:

$$\int_V (\boldsymbol{x} - x_1)^{\boldsymbol{p}'} \, dV \tag{16}$$

for $\boldsymbol{p}' = 0, 1, 2 \cdots \boldsymbol{p}$. Given that we are requested to calculate moments with the origin at $\boldsymbol{x}_0$, we identify a point $\boldsymbol{x}_1$ by the following procedure.

Let E denote the set of points lying on the edges of V. Let I denote the set of points where the interface intersects the edges in E. That is, $I = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\} \cap E$. We define a point, denoted earlier x_{EB}, as the average of points in I. If we denote the number of points in I by N, then the equation for x_{EB} is:

$$x_{EB} = \frac{1}{N} \sum_{x \in I} x \tag{17}$$

Given that we wish to compute a moment with respect to an origin at a point $\boldsymbol{x}_0$ as in equation 15, we perform this work in two steps: first we calculate all moments of degree less than or equal to $\boldsymbol{p}$ with respect to an origin a $\boldsymbol{x}_1 = x_{EB}$ using an Taylor approximation of the normal also centered an x_{EB}. Secondly, we use the binomial theorem to represent the desired moment.

This extra step diminishes the error, although the order of accuracy remains the same (quote example). We attribute the improved accuracy to two factors. First the error in the Taylor approximation to the normal is diminished, provided $\boldsymbol{x}_{EB}$ is closer to the interface on average than $\boldsymbol{x}_0$. Secondly, empirically, there are fewer examples where different components of the solution vector vary widely in magnitude.

7 Refinement

There are two cases in which the algorithm can provide an inadequate calculation.

Some implicit functions, arising for example from digital elevation maps or image data, have features at every resolution. A calculation at any fixed h will come under stress at regions of high curvature, when edges have multiple intersections, and when thin bodies are marginally resolved.

On the other hand, even well-resolved, slowly varying implicit functions, such as a sphere with a radius much larger than h, can lead to inaccurate calculations for lower dimensional problems. For example, given a sphere in three dimensions, there are cells, V, where the interface is nearly tangent to a face of V. In an extreme case, the projection of the interface onto a two dimensional face results in a closed curve entirely contained within the face. Such a calculation is clearly under-resolved, leading to inaccurate answers for the face moments and an inaccurate right hand side for the three dimensional moments.

The solution we employ for this problem requires that we detect when this phenomena occurs. Having detected that refinement is necessary,

- We partition the cell into 2D smaller problems.
- Since the moments of the smaller problems are with respect to an origin appropriate for the smaller problem, we employ the binomial theorem to combine this information into moments with respect to the origin of the larger problem.
- Having calculated moments with respect to the correct origin over smaller disjoint regions, we sum the moments to achieve the final answer.

It remains to decide what method works best for detecting that refinement is necessary. In some cases, for example in implicit functions arising from digital elevation maps or image date, a priori knowledge of the the location of thin bodies can lead to a simple method of tagging certain cells. Otherwise, for example in implicit functions arising from analytic expressions, a priori bounds on the derivatives of the implicit function may be available. These bounds may be used to anticipate when the change in the normal over the interface exceeds some threshold. Similarly, direct discrete estimates of the change in the normal can be calculated.

The least squares part of the alorithm can be asked to reprot when the constraints were active, which can be used to infer that the calculation is under resolved. The least squares module can also report the residual, which can contribute to a refinement criterion.

All or some of the methods above can be combined in a boolean algebra to create complex criteria.

8 Data Structures

Our implementation depends on two data structures, that we call IFData and CutCellMoments. In our C++ implementation, these are templated classes that use the number of space dimensions as the template parameter. For the purposes of exposition, it suffices define to data structures parameterized by an integer parameter, dimension. Thus we may write IFData(dim) or CutCellMomentData(dim) when we wish to show the dependence on the parameter. For example, moments over $\mathcal{V} \subset \mathcal{R}^D$ are stored in a CutCellMoments(D).

8.0.1 IFData

IFData(d) is constructed from a cell center, a grid spacing, and a dimension. It encapsulates relevant information in ϕ used to construct the linear system.

- At each of the 2^d corners of the rectangular cell, we record whether the corner is in Ω, the complement of Ω, or Ω_0, which is nearly always determined by evaluating ϕ at the corner. However, for ϕ sufficently close to 0, we assign the corner to Ω_0. In our work we set the corner value to 0 when $|\phi| < 1.0$ e-15. This rare, slight modification of ϕ increases the robustness of algorithms that make logical choices based on whether an edge is completely contained in Ω or completely contained in the complement of Ω.
- At each edge of the rectangular cell, we record an intersection point if the interface intersects the edge. We discover this intersection through a root-finding algorithm such as Brent's algorithm.
- We record the average of the intersection points and denote this point, the "local" origin.
- Another point, passed in on construction of the IFData(d), is denoted "parent".
- At the "local" origin, we evaluate ϕ, $\boldsymbol{n}$ and a pre-determined number of partial derivatives of $\boldsymbol{n}$.

8.0.2 CutCellMomentData

A CutCellMomentData is contstructed from an IFData. For any space dimension d, such that $2 \leq d \leq D$, CutCellMoments(d) contains

- an IFData(d)
- moments over regions, $\mathcal{V} \subset \mathcal{R}^d$
- moments over regions $\mathcal{V} \cap \Omega_0 \subset \mathcal{R}^{d-1}$
- CutCellMomentsData(d-1) corresponding to the coordinate aligned faces of $\mathcal{V}$

A CutCellMomentData(1) contains integrals of monomials over a line segment.

Fig. 1. Without refinement, The volume of the sphere does not converge in the max norm.

Fig. 2. The lower dimensional problem is under-resolved when a coordinate face is tangent to the interface.

Fig. 3. Convergence to second order for the sphere after refining at six cells.

Fig. 4. In some cases, a sufficiently accurate answer is non-physical.

Fig. 5. Using constraints, centroids are within the cell and volumes are positive.

9 Results

9.1 Three dimensional, second order results

Our first result illustrates the need for refinement. Figure (1) shows an absence of the specified convergence rate in the max norm for the well-resolved sphere. Closer inspection reveals that in the six places where a face of a computational cell is nearly tangent to a coordinate plane, the associated two dimensional problem is under-resolved. See figure (2) We identify under-resolved cells based on several criteria, such as the change in the normal, whether constraints were active, or the size of the residual. When a cell is identified as under-resolved, we calculate moments on 2^D finer problems, which are then recombined for the final calculation. Figure (3) shows the results with refinement.

Our second result illustrates the use of constraints to keep computed answers physically realistic. In figure (4), we demonstrate that in typical applications, one inexpensive solution to under-resolution, can be the use of constrained least squares. Use of contraints can keep the answer from violating physical constraints without the additional work of refinement. In addition the use of constraints can incorporate application specific information, such as the “no-undercut” assumption in the SF Bay and Delta shoreline.

Fig. 6. Slices of the error for the fouth order calculation on the sphere

Fig. 7. Using more terms in the Taylor series for the normal, we exhibit fourth order convergence on the sphere.

Fig. 8. Using Richardson extrapolation we exhibit fourth order convergence for a polynomial implicit function: $2x^2 + 4y^3 + 3xy$

Fig. 9. Third order convergence for a three dimensional sphere moving in time.

9.2 Three dimensional, fourth order results

We exhibit fourth order results for a sphere. Figure (6) shows slices of the error. Figure (7) shows the convergence rate. In figure (8) Richardson extrapolation illustrates fourth order convergence using a polynomial implicit function.

9.3 Four dimensional, third order results

In figure (9), we exhibit third order convergence on three dimensional sphere moving in fourth dimension, time.

10 Conclusion

It has been shown that moment calculations can be done in arbitrary dimension to arbitrary accuracy. If the irregular boundary is defined by implicit functions, computing these moments can be computed using the divergence theorem, Taylor expansions, least squares, recursion, and 1D root finding. Where the interface is under-resolved, automatic detection leads to a refined calculation. For grid generation, the resulting computations are embarassingly parallel, robust and efficient over complex domains arising from data, man-made structures, and analytic representations.

This research was supported by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231.

References

[GI83] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex convex quadratic programs. *Math Program*, 27:1–33, 1983.