Journal of Physics: Conference Series **125** (2008) 012084 doi:10.1088/1742-6596/125/1/012084

Embedded boundary algorithms and software for partial differential equations

Phillip Colella, Dan Graves, Terry Ligocki, David Trebotich[1] and Brian Van Straalen

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA [1]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

E-mail: bvstraalen@lbl.gov

Abstract. In this paper, we give an overview of a set of methods being developed for solving classical PDEs in irregular geometries, or in the presence of free boundaries. In this approach, the irregular geometry is represented on a rectangular grid by specifying the intersection of each grid cell with the region on one or the other side of the boundary. This leads to a natural conservative discretization of the solution to the PDE on either side of the boundary. Stable and robust hyperbolic and linear elliptic/parabolic solvers have been designed and implemented. Example applications of this approach are shown for compressible and incompressible gas dynamics problems in complex geometries, and for surface diffusion in a cell membrane.

1. Introduction

Structured grid numerical methods based on the finite-volume approach have a number of significant computational and numerical advantages: regular predictable memory access, implicit structure (thus saving the memory and computation required for data interpretation), higher accuracy for less computation due to stencil error term cancellation, fast and robust geometric multigrid solvers, and efficient extensions to block-structured locally refined grids to provide multiresolution capabilities. To provide comparable simulation capabilities in the presence of complex geometry, we use the embedded boundary (EB) approach. Domains with irregular boundaries are represented by the intersection of the domain with Cartesian cells, thus forming irregular control volume(s). Partial differential operators in conservation form are then approximated by using a finite-volume discretization of the divergence operator, i.e. as a sum of fluxes through faces. For hyperbolic PDEs the fluxes are functions of the conserved variables, while for elliptic/parabolic PDEs they are proportional to the gradient of the solution. Both take as the basis of their discretization

$$\frac{1}{V}\int_V \nabla\cdot\vec{F} = \frac{1}{\kappa h}(\sum_i \pm\alpha_i F^i) + \alpha_B \vec{F}^B\cdot\vec{n} \qquad (1)$$

(figure 1, left). Fluxes on faces that have area fractions α that are not equal to unity are constructed by interpolating fluxes centered at the centers of the Cartesian faces, shown in 2D in figure 1, right. The figures in 1 illustrates the kinds of geometric data we need to compute accurate simulations: boundary centroid, boundary normal $\vec{n}$, face centroids: $[\bar{x},\bar{y},\bar{z}]$, aperture

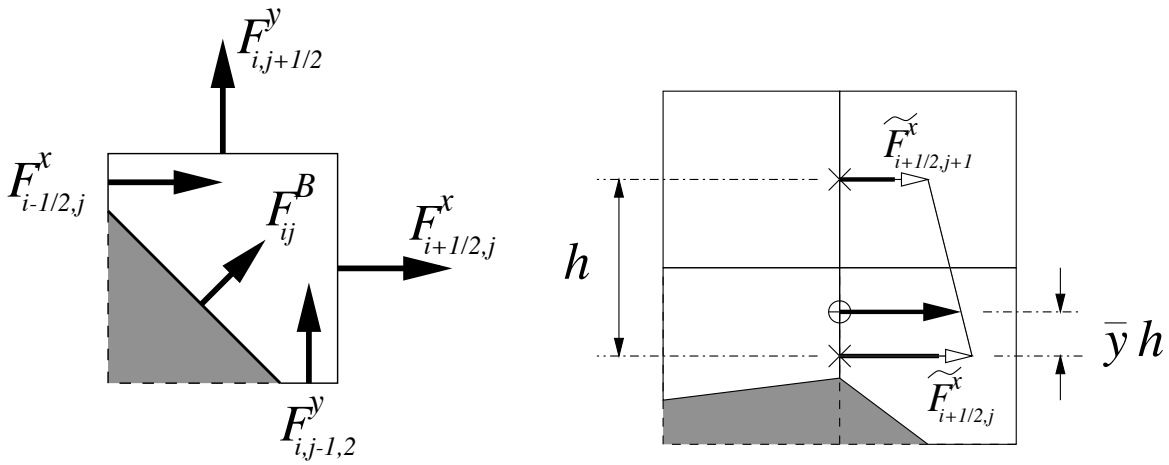


Figure 1. Left: fluxes that make up a conservative update of an irregular cell. Right: constructing a non-unit area fraction aperture flux by interpolation.

area fractions: α, volume fractions: κ. The computation of these quantities is the analogue of mesh generation for body-fitted grid methods.

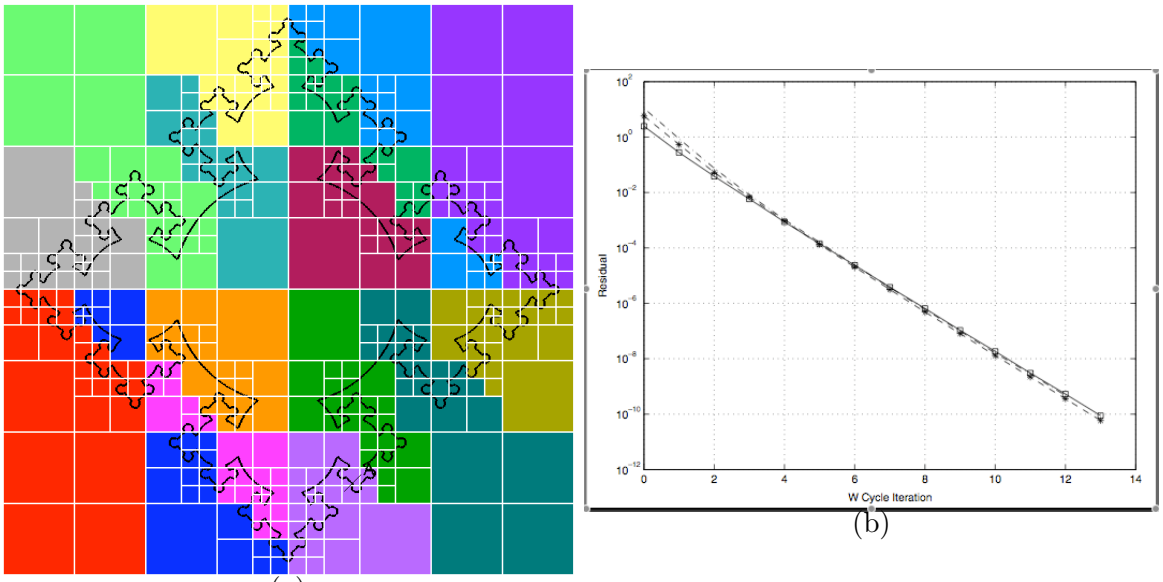
2. Mesh generation

The volume centroids, volume fractions, area centroids, area fractions, and boundary normal required in (1) can be extracted from surface triangulations through careful application of computational geometry in a very general fashion (as done in [5]). We have developed an alternative approach, based on representing the irregular domain Ω in terms of an implicit function $\phi : \mathbb{R}^D \rightarrow \mathbb{R}$

$$\Omega = \{\boldsymbol{x} : \phi(\boldsymbol{x}) < 0\} \ , \ \partial\Omega = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\}. \tag{2}$$

Implicit functions are standard representations for image data, where they are constructed using level-set methods for segmentation of discrete representations of images; and in geophysical data, in the form of digital elevation maps. It is also straghtforward to build complex domains out of simpler ones using composition. Given two domains represented by implicit functions ϕ_1 and ϕ_2, then the union of these domains is represented by the implicit function $\phi = min(\phi_1, \phi_2)$, and the intersection by $\phi = max(\phi_1, \phi_2)$. In addition, the EB mesh generation problem for domains defined by implicit functions can be reduced, by means of repeated applications of the divergence theorem, to finding the intersection of coordinate lines with the zero set of ϕ, together with a set of well-conditioned linear least-squares problems for the moments for all higher dimensions. The details of this algorithm are described elsewhere in these proceedings [2].

In addition to the real-valued quantities associated with each control volume and cell face, we also need to compute and store the topological information describing the entire collection of control volumes and their connectivity. Every $[i, j, k]$ Cartesian cell in the index space is classified as either *Inside*, *Outside* or *Irregular*. Cells *Outside* have have a positive value for the the implicit function everywhere and are outside the computational domain. The opposite holds for all *Inside* cells. *Inside* cells have unit volume fraction and unit aperture area fractions and centroids on the zero axis. In *Irregular* cells the implicit function change sign. In a similar fashion, boxes in the Cartesian index space $[i, j, k]_{low}, [i, j, k]_{high}$ can also be classified as *Outside* if every cell in the region is outside, *Inside* if every cell is inside, or *Irregular* if any cell in the region is irregular.



(a) (b)

Figure 2. (a) `EBIndexSpace` database object. Colors represent processor assignments. White represents box boundaries. Black represents the zero of the implicit function. The implicit function in this case is a composite solid geometry fractal design created from recursive `unions` of spheres. (b) Plot of the max norm of the partial volume-weighted residual versus W-cycle iteration. The graphs are for meshes of size 64 (□), 128 (*), and 256(-)

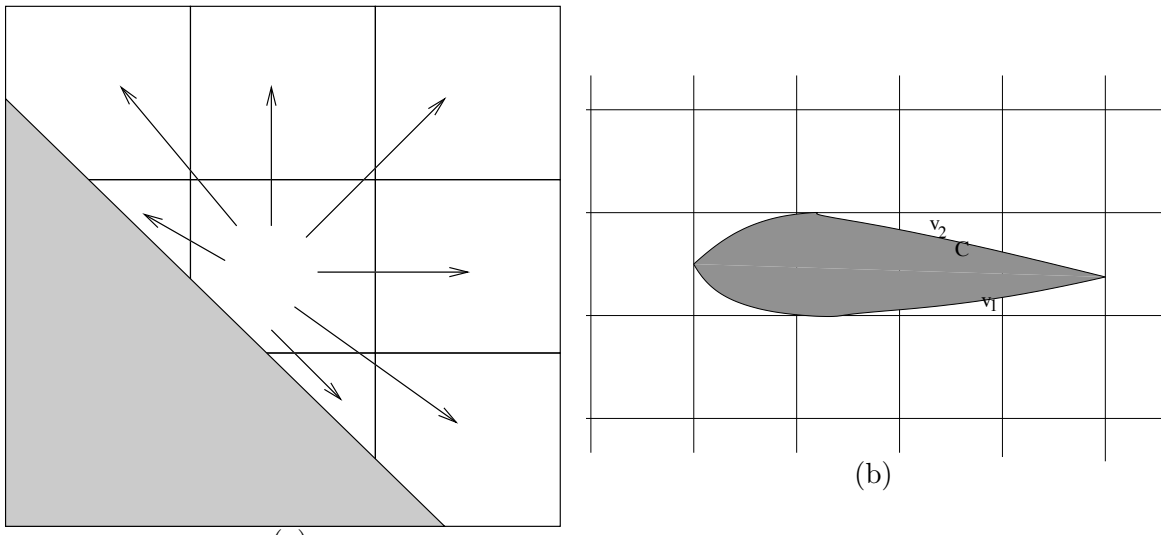
3. Parallel Computation

Embedded boundary methods admit a straightforward extension of the domain-decomposition approach used in other structured-grid and block-structured AMR methods [6]. The computational domain at a given level of refinement is covered by a collection of boxes in the Cartesian index space, each of which is assigned to a processor. For each box, one can define the corresponding EB data. Then the organization of parallel computation is the same as for the non-EB case, for example, applying operators by iterating over the boxes on each processor, exchanging ghost cell values.

Chombo does not compute the geometric moments on-the-fly, but builds a distributed database based on a recursive bisection parallel decomposition of the Cartesian index space. Starting with the overall problem domain we query the implicit function if the box covering the domain is *Inside/Outside/Irregular*. In a typical complex geometry problem this will be *Irregular*. The box is divided into into two equal subboxes, and the algorithm recurses into each subbox. The recursion stops when a region is either *Inside* or *Outside* or a minimum box size is reached.

This algorithm creates a covering set of the domain. A space-filling curve is constructed for this collection. These boxes are then assigned to processors with a heuristic load-balancing algorithm. Each processor then visits each of its *Irregular* boxes and proceeds to query each $[i, j, k]$ cell in the box and make the *Inside/Outside/Irregular* query. *Irregular* cells have their geometry moments computed and stored. A sample distribution to processors, shown by colors, is given in figure 2a. *Inside* and *Outside* boxes are represented sparsely, with just an integer flag. *Irregular* boxes are stored as a dense mask of cell flags. *Irregular* cell data is stored in auxiliary sparse data structures. This then forms the `EBIndexSpace` object.

The data distribution for the EB grid generation is not same as for the simulation unknowns. After each regridding operation in an AMR simulation there is a geometry data localization operation. Each level of grids has an associated `EBISLayout` that represents a locally cached



(a) (b)

Figure 3. (a) Redistribution of nonconservative update to neighboring cells (b) Coarsened `EBIndexSpace` showing a multivalued cell

duplication of the geometric data for the grids that are on-processor.

4. Solvers

In the applications being developed using the EB algorithms and software, we have typically used semi-implicit methods. These are based on expressing the governing partial differential equations are as a decomposition in terms of operators of classical type, i.e. *hyperbolic*, *elliptic* and *parabolic*. Explicit methods are used for discretizing hyperbolic operators, while fast linear solvers are required for elliptic PDEs and for implicit discretizations of parabolic terms. These are combined with implicit methods for advancing parabolic systems and implicit solvers for elliptic constraints. The overall algorithms are robust and efficient with stability properties that are insensitive to the resolution of the geometric details.

4.1. Hyperbolic Solvers

For hyperbolic operators, we are solving

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} = 0. \tag{3}$$

We would like to discretize (3) in time using an explicit method, with (1) for the spatial discretization. As written there is a severe restriction on the explicit time step because the volume fraction κ can be arbitrarily small. This is the "small cell problem" that appears in most cut/embedded cell methods. The approach we have taken [9] is to compute two different approximations of $\nabla \cdot F$: a nonconservative approximation that ignores the embedded boundary and is known to be stable for the regular grid spacing CFL condition$(\nabla \cdot F)^{NC}$, and a conservative but possibly unstable approximation $(\nabla \cdot F)^{C}$. A lineaqr combination of these fluxes is a stable discretization, but adding a conservative and nonconservative update results in a nonconservative update. To recover conservation there is a redistribution of the nonconservative update to neighboring cells (see figure 3a).

To compute the stable nonconservative update can require the computation of a flux on an aperture that is entirely outside the simulation domain. The details are complicated but involve the existence of smooth extensions of solution derivatives normal to the boundary face.

4.2. Elliptic Solvers

One of the principal advantages of Cartesian grid discretizations of elliptic PDEs is the efficacy of geometric multigrid in providing algorithmically efficient, high-performance solvers [4]. This advantage persists in the case of EB discretizations(fig. 2b), but leads to a more complicated implementation [7].

A key step in multigrid algorithms is *coarsening*. To restrict the residual to the next multigrid level, one performs geometric averaging of the grid by a small interger factor, usually two. In the non-EB case, computing the relationship between the locations of the coarse and fine data involves simple integer arithmetic. In the EB case, both the data access and the averaging operations are more complicated. Coarser geometry moments are generated by sums and averages of the finest geometry values. The volume fraction in a coarse cell must be the sum of the volume fractions of the finer covering cells, similar for area apertures, and so on. This prevents spurious residual sources during restriction and prolongation. It is also essential that coarsening a geometry preserves the topology of the finer EB representation. The consequence of this requirement is that geometric features at coarser levels of the `EBIndexSpace` will have a length scale that is smaller the grid spacing, which leads to the "thin body problem". The field needs to maintain separate smooth extensions for each boundary face. Thus the cell shown in figure 3b has more than one value. These multicells must be preserved and not merged during coarsening. Merging these cells results in multigrid that does not converge because the coarser level generates corrections that do not respect the finer level's boundary conditions. `EBIndexSpace` must maintain these cells in auxiliary sparse data structures at each level of multigrid refinement. All this means the `EBIndexSpace` needs to maintain successively coarsened representations of the geometry as part of the database.

5. Results

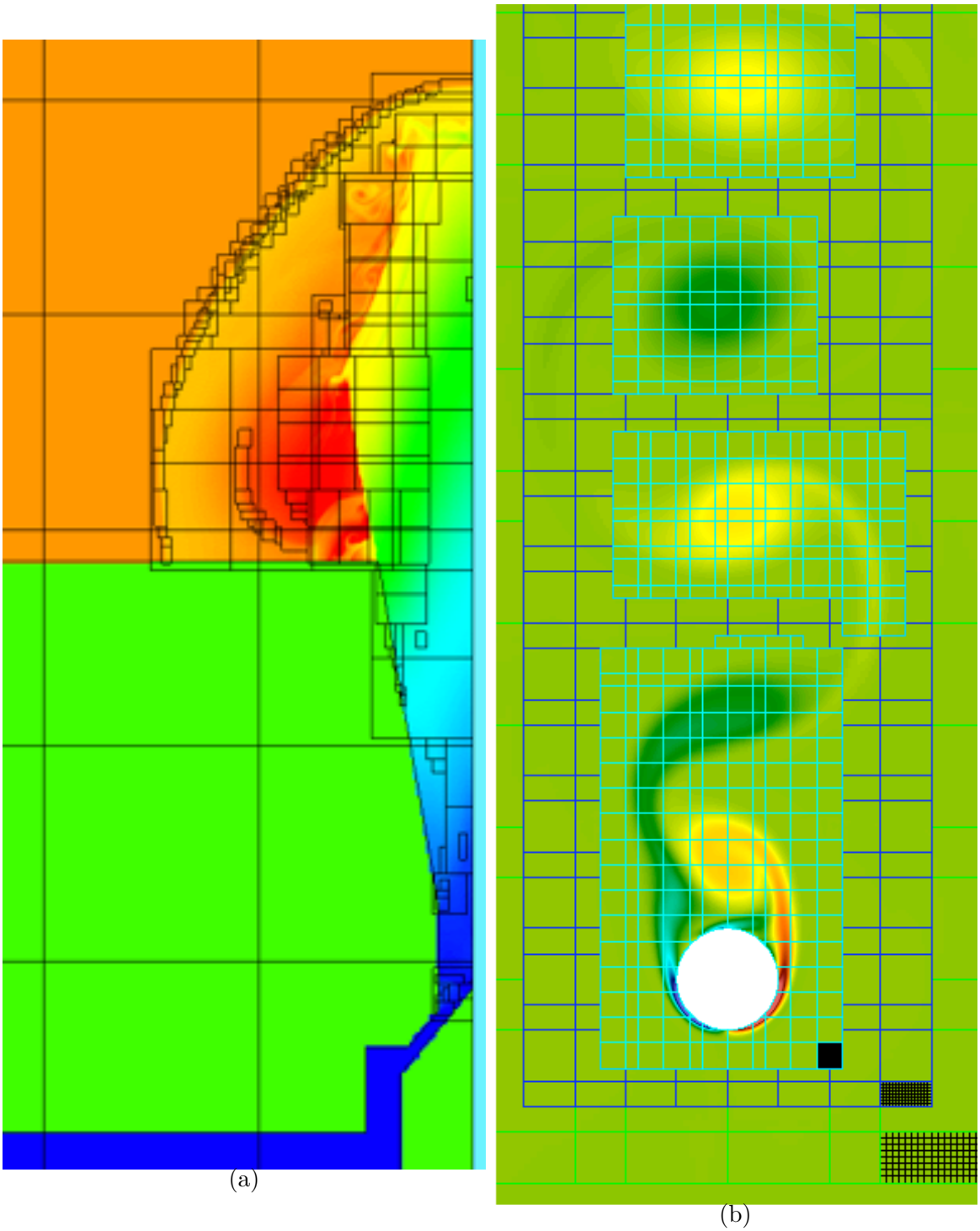
In figure 4a, we show an inviscid gas dynamics calculation solved using the explicit EB method based on a second-order unsplit Godunov scheme [9], in r-z coordinates. This problem is motivated by the problem of gas jet formation in novel laser wakefield accelerator designs.

In figure 4b, we show a calculation of a viscous incompressible flow problem using a semi-implicit projection method. The hyperbolic advective terms are computed using a version of the method in [9]. These are then used as a right-hand side for a second-order implicit Runge-Kutta scheme [10] for the parabolic viscous terms to create a second-order accurate stable approximation for the viscous forces, and requiring the solution of a Helmholtz equation. Finally, an approximate update for the velocity field obtained from the incrementing the velocity at the old time by the advective and viscous terms is projected to extract its divergence-free part, requiring the solution of a Poisson equation. There are additional operations that appear in the non-EB algorithms for supporting the use of AMR, similar to those described in [12]. This algorithm permits us to take time steps that are only limited by the CFL condition $\triangle t = O(h/v_{\max})$.

Finally, we show an example that illustrates the value of an implicit function representation of the geometry. In this case, microscopy data is turned into signed distance function of a cell membrane. This allows us to generate an `EBIndexSpace` corresponding to an annular region of fixed thickness around the membrane. If the annulus has a thickness on the order of the grid spacing h, then applying homogeneous Neumann boundary conditions on the surfaces of this region and solving the 3D parabolic diffusion equation results in an $\mathrm{O}(h^2)$ approximation of diffusion within the membrane [13].

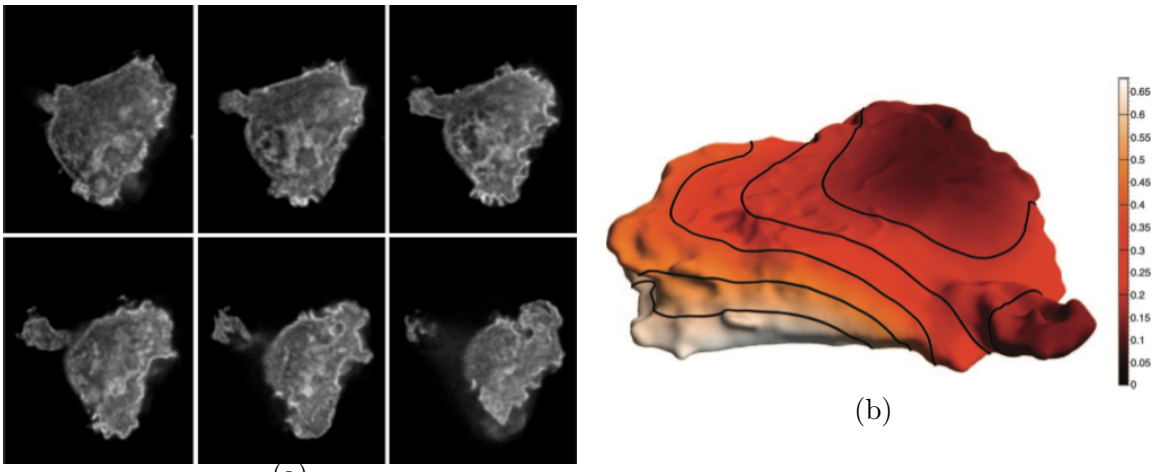
6. Future work

The embedded boundary approach can be extended to all regimes of existing APDEC simulation capabilities. The first major new technology will be the use of the *Allspeed* approach



(a) (b)

Figure 4. (a) Compressible gas dynamics flow from high pressure reservoir into a vacuum through a nozzle. (b) Kármán vortex street behind cylinder at Re=300. Effective fine grid resolution 2048x4096. Color scaled for magnitude of vorticity [-175, 175]. Images made with the VisIt visualization package.



(a) (b)

Figure 5. (a) 2D slices through a gray-scale image of an hl-60 cell obtained by using deconvolution microscopy. Image is ≈2,000 times actual size. (b) Solution for surface diffusion at time = 50 s. The time step is 5.0 s.

to compressible flows with significant acoustic energy modes [14]. Originally developed to handle transonic flows, the APDEC center is now using this approach to handle a range of multiwavescale physics.

With *Allspeed* in place we will be able to bring together the combustion modeling effort in APDEC with the embedded boundary approach and start performing full engineering device combustion modeling with full chemistry and at resolutions that do not utilize sub-grid models.

The EB boundary can also represent a defined or tracked interface. Development of a multifluid modeling capability is underway, as well as basic research into using EB for moving interfaces A movinf interface creates higher dimension space-time apertures, which are easily handled with the divergence theorem concepts.

Embedded Boundary Chombo, an open-source software release containing all the features and examples shown in this paper, is expected by the end of FY08.

Acknowledgment

Research supported by the Office of Advanced Scientific Computing Research of the US Department of Energy at the Lawrence Berkeley National Laboratory under contract number DE-AC02-05CH1123, and at the Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.

References

[1] Colella P 2001 Volume-of-fluid methods for partial differential equations *Godunov Methods: Theory and Applications* (E F Toro) 161–177
[2] Ligocki T, Schwartz P, Percelay J and Colella P 2008 Embedded boundary grid generation using the divergence theorem, implicit functions, and constructive solid geometry *Proc. SciDAC 2008*
[3] Martin D F, Colella P and Graves D T 2006 A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions (LBNL report LBNL-62025)
[4] Martin D F and Colella P 1996 Solving Poisson's equation using adaptive mesh refinement (UC Berkeley Electronics Research Laboratory report No. UCB/ERL M96/66)
[5] Aftosmis M, Berger M J and Melton J 1998 Robust and efficient Cartesian mesh generation for component-based geometry *AIAA J.* **36**(6):952–960
[6] Colella P, Bell J, Keen N, Ligocki T, Lijewski M and Van Straalen B 2007 Performance and scaling of locally-structured grid methods for partial differential equations *Proc. SciDAC 2007*

[7] Schwartz P, Barad M, Colella P and Ligocki T J 2006 A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions (LBNL-56607) *J. Comput. Phys.* **211**:531–550
[8] Colella P, Graves D, Van Straalen B and Trebotich D 2008 Performance of embedded boundary methods for CFD with complex geometry *Proc. SciDAC 2008*
[9] Colella P, Graves D T, Keen B J and Modiano D 2006 A Cartesian grid embedded boundary method for hyperbolic conservation laws (LBNL-56420) *J. Comput. Phys.* **211**:347–366
[10] Twizell E H, Gumel A B and Arigu M A 1996 Second-order, l0-stable methods for the heat equation with time-dependent boundary conditions *Advances in Computational Mathmatics* **6:**333–352
[11] Colella P and Woodward P 1984 The piecewise parabolic method (PPM) for gas-dynamical simulations *J. Comp. Phys.* **54**(1):174–201
[12] Martin D and Colella P 2000 A cell-centered adaptive projection method for the incompressible Euler equations *J. Comput. Phys.* **163**:271–312
[13] Schwartz P, Adalsteinsson D, Colella P, Arkin A and Onsum M 2005 Numerical computation of diffusion on a surface *Proc. Nat. Acad. Sci.* **102**:11151–11156 1999
[14] Colella P and Pao K 1999 A projection method for low speed flows *J. Comput. Phys.* **149**:245–269