

The Applied Partial Differential Equations SciDAC  
Integrated Software Infrastructure Center (APDEC)  
Math ISIC Program Review, 5/13-14/2003

**Lead PI:**

P. Colella, LBNL

**Principal Investigators:**

J. Bell, LBNL

D. Brown, LLNL

M. Berger, New York Univ.

R. Leveque, Univ. of Washington

M. Minion, Univ. of North Carolina

G. Puckett, Univ. of California, Davis

C. Rutland, Univ. of Wisconsin

R. Samtaney, PPPL

April 20, 2003

<http://davis.lbl.gov/apdec/>

# 1 Introduction

Many important DOE applications can be described mathematically as solutions to partial differential equations exhibiting behavior on multiple length scales. Combustion for energy and transportation is dominated by the interaction of fluid dynamics and chemistry in localized flame fronts. Fueling of magnetic fusion devices involves the dispersion of material from small injected fuel pellets. The successful design of particle accelerators relies critically on the confinement of the charged beams to a small subset of the total volume.

In this project, we are developing a new class of simulation tools for these and other multi-scale problems. The algorithmic approach we are taking, discussed in detail in section 3.2 of the original proposal, is based on the use of finite-difference methods on structured grids combined with block-structured adaptive mesh refinement (AMR) to represent multi-scale behavior. In this approach, the physical variables are discretized on a spatial grid consisting of nested rectangles of varying spatial resolution, organized into blocks. This hierarchical discretization of space can adapt to changes in the solution to maintain a uniform level of accuracy throughout the simulation. We also can vary the temporal resolution to match the spatial resolution.

The use of AMR requires the consideration of new mathematical, algorithmic, and software issues in order to represent the coupling between different scales. For that reason, we have taken an end-to-end approach, developing self-contained new simulation capabilities based on AMR. These include AMR fluid simulation codes for turbulent combustion in laboratory-scale flames and for non-ideal magnetohydrodynamics problems arising in magnetic fusion; AMR-PIC codes for computing particle-in-cell space charge effects for beam dynamics in accelerator design problems; and an AMR embedded boundary code for simulating gas jets in laser-driven plasma-wakefield accelerators. The progress we have made in developing these applications is described in section 2 of this document. Although these applications are quite diverse physically, there is considerable overlap in their mathematical structure. We have developed a component design for our software based on that common mathematical structure to maximize reuse across applications, as well as portability across platforms. We describe the status of the software development effort in section 3 of this document. In section 4 of this document we describe the status of the advanced algorithm development efforts. The final section contains a summary of the staffing and budget.

## 2 Applications Development

### 2.1 Overview

There are three applications areas for which the APDEC project is developing AMR algorithms and software: combustion, magnetic fusion, and accelerator design. For each of these areas, there is an obvious need for multiscale simulation capabilities, which is discussed in detail in section 3.1 of the original proposal. Furthermore, to the extent that all of these problems can be represented mathematically as solutions to classical partial differential equations, we expect to be able to develop a common software infrastructure to support all of them. However, the level of maturity of adaptive methods for these areas varies widely.

AMR for combustion has been the focus of a sustained research effort on the part of the investigators for a decade or more. Consequently, the work described here is focused on the production of new capabilities that lead immediately to new scientific results. At the other extreme, both projects in accelerator design - AMR-PIC for beam dynamics calculations, and gas-jet simulations for laser-driven plasma-wakefield accelerators - require capabilities for which the algorithm development is of much more recent provenance, and whose development as software was started at the beginning of the APDEC project. For that reason, we are just now reaching the point where we have initial implementations of the code capabilities required to simulate the problems in that area. The third application, magnetic fusion, is intermediate between the two. While there is little prior art in adaptive methods for nonideal MHD, the problem is sufficiently similar to the fluids problems for which our adaptive techniques are quite mature that there was software and algorithmic expertise that we could leverage.

The differing levels of maturity also have an impact on the relationship between the software development effort and the application efforts. For magnetic fusion and accelerator modeling, the application efforts are heavily dependent on the software development effort efforts described in section 3 of this document. We recognized this from the outset, and the priorities in the software development process have been focused on producing the required capabilities in a timely fashion, and on supporting the applications developers. In the combustion application, there is already in place a considerable software infrastructure based on the BoxLib library, the predecessor to the Chombo library that is the basis of the current APDEC software development effort. Consequently, there has been no need for the software development team to support the combustion effort. We expect that this will change when the combustion effort requires the embedded boundary capabilities being developed in Chombo for problems with complex geometries.

Finally, we wish to emphasize that the principal focus of the APDEC project is on applications. Ultimately, our goal is to provide new simulation capabilities to these communities that will enable them to investigate scientific problems that were previously not accessible. This is reflected in the budget for the project: each application area has 1 FTE funded directly by APDEC for the development of that application. We discuss in further detail the staffing and budget issues in section 5 of this document.

## **2.2 Combustion**

### **2.2.1 Goals and Approach**

Combustion is one of the cornerstone applications for the APDEC ISIC. Our goal in this application to develop and apply new simulation methodologies on high-performance parallel architecture to understand the interplay between chemistry and fluid dynamics in flames. The focus of our work is on studies combining detailed chemistry and transport where the fluid-dynamics is modeled directly without incorporating models for turbulence or turbulence / chemistry interaction. Our core methodology uses a low Mach number adaptive projection formulation that provides a substantial improvement in computational efficiency compared to standard compressible DNS methods for combustion.

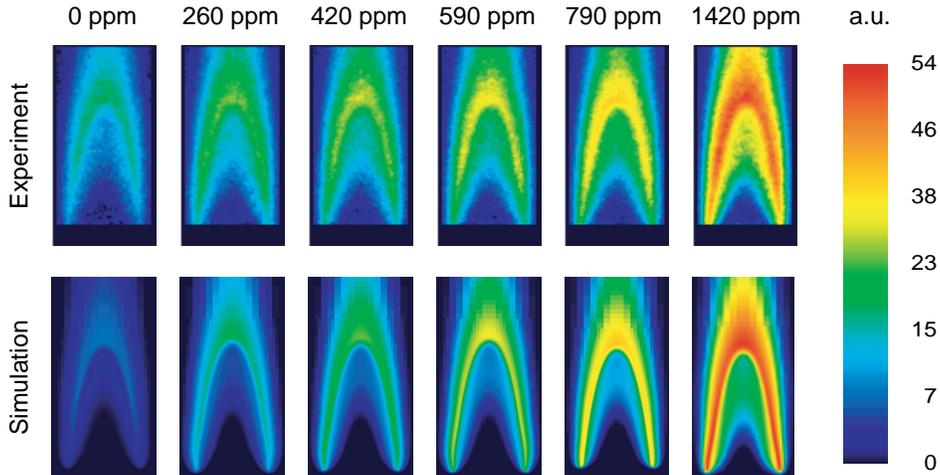


Figure 1: *NO A-X(0,2) excitation LIF images obtained (top) from measurement and (bottom) by synthetically processing the results of the flame simulation, for different  $\text{NH}_3$  seeding concentrations. The experimental data and the synthetic LIF intensities are prepared independently; the same scale applies to both.*

### 2.2.2 Accomplishments

Our research in combustion thus far has focused on two target investigations: pollutant production in steady ammonia-seeded methane diffusion flames, and turbulence-chemistry interactions in wrinkled premixed methane flames.

We are investigating ammonia-seeded diffusion flames in collaboration with researchers at the University of Heidelberg and the Technical University of Denmark. The goal of this work is to improve the understanding of NO pollutant formation due to fuel-bound nitrogen characteristic of bio-mass fuels. For these problems, data is gathered from the experiment using laser-induced fluorescence (LIF). In the “two-line” analysis approach for temperature extraction, two frequencies associated with the NO molecule are excited by the laser over a two-dimensional sheet through the vertical midplane of the flame, and the resulting LIF signals are processed using the known temperature-dependence of the bands. The absolute NO concentration can be inferred from either signal if the temperature and mole fractions are known for all species that may quench the NO LIF signal. Unfortunately, the NO signal is strongly quenched by the  $\text{O}_2$  molecule, so that the two-line approach depends critically on assumed chemical profiles.

We have developed a new approach that more closely intertwines simulation and experimental data interpretation. Using an two-dimensional axisymmetric model, we computed steady diffusion flame solutions corresponding to the experimental setup. After validating the computational results against the temperature profiles inferred from the experiment, we used the simulation results, combined with quantum-dynamical quenching models to generate *numerical LIF* images. Across the range of experimental parameters, these numerical LIF images showed exceptional agreement with the raw LIF data from the experiment. A comparison of the experimentally-measure and the synthetic computational LIF is presented in Figure 1.

Additionally, using simulated chemical distributions directly in the quench calculations, we demonstrated very good agreement between the computed and experimentally inferred NO concentrations [4].

In our second combustion application, we applied the three-dimensional version of our numerical algorithm technology to understand the detailed response of a lean premixed methane flame sheet to turbulence in the unburned mixture. These simulations are the largest of their type ever attempted, and were achievable through our use of locally-adaptive gridding methods coupled to the low Mach number formulation. The flame chemistry is modeled with a 19 species, 84 reaction subset of the GRIMech-1.2 methane chemistry mechanism. The calculations are carried out over the  $8 \times 8 \times 16$  mm domain on a hierarchical grid structure with an effective resolution of  $256 \times 256 \times 512$  cells. We considered two different cases corresponding to different levels of turbulent intensity. Flame sheet images from these simulations are depicted in Figure 2.

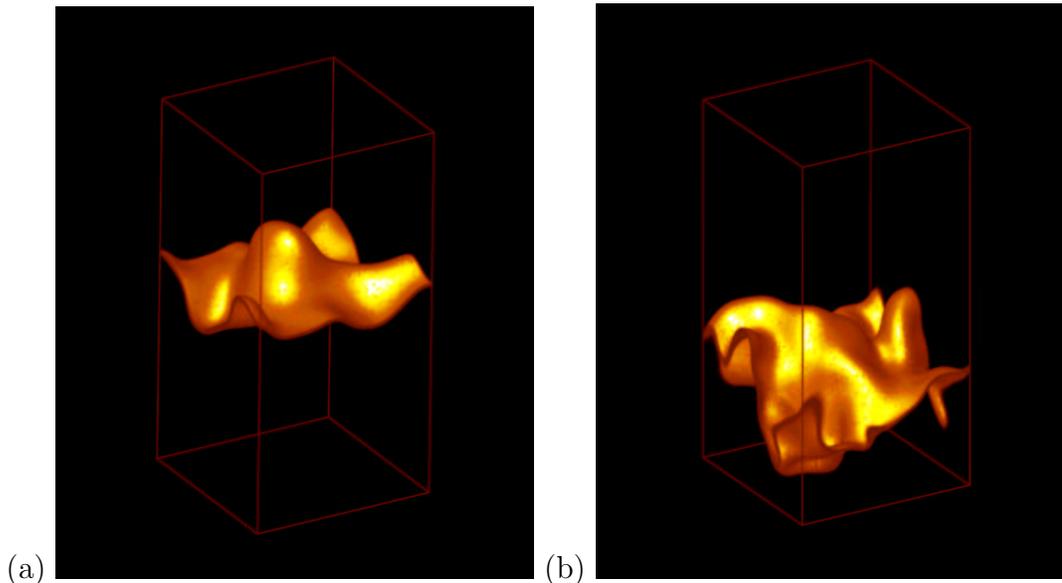


Figure 2: *Volume rendered image showing surface of maximum heat release for the weak (a) and strong (b) turbulence cases.*

Simulation results confirm that wrinkling due to turbulence is the dominant factor leading to an increased effective flame propagation speed. Under the turbulence regimes studied thus far, the increase in speed is predominately attributable to the increased flame surface area. This suggests that the time-dependent stretch and flame curvature have little aggregate effect on the overall flame structure in the cases investigated. However, when we examine local details of the flame, we find a different picture. Even for low turbulence levels, the heat release in the flame correlates strongly with flame curvature. In fact, the strength of many of the reactions in the system are correlated to flame curvature, an observation which is likely related to focusing and defocusing preferential hydrogen diffusion effects. We found that an additional effect of the turbulence was to move apart the regions of production and destruction for a subset of the chemical species, particularly in regions of negative curvature. As a result, the residence times, and computed molar concentrations of these species were

correlated to the curvature as well [6].

We are currently extending that work on turbulent premixed combustion to modeling of a laboratory-scale premixed turbulent flame. The configuration for the experiment we are considering consists of a thin rod across the center of a nozzle with a partial obstruction ten centimeters upstream of the rod to generate turbulence. The presence of the rod creates a local stagnation in the flow which leads to the formation of a V-flame. Simulations for this configuration specify flow at the exit of the nozzle using a characterization determined from the experimental data and use that data as boundary conditions for our low Mach number combustion algorithm. As in our earlier work on the turbulent flame sheet, we model the methane chemistry using DRM-19 and we model transport using a mixture model for differential diffusion. Initial simulations show that we can successfully predict a stable turbulent V-flame. Furthermore, mean statistical properties of the flame brush match the experimental data. Comparison of the simulation, presented in Figure 3 with particle-induced-velocimetry (PIC) data show that our simulations also do a good job of predicting the basic flame morphology. In these images, the spreading angle of the “V” is determined by the turbulent enhancement of the flame speed. The agreement between experiment and computation shows that we are accurately capturing the flame dynamics.

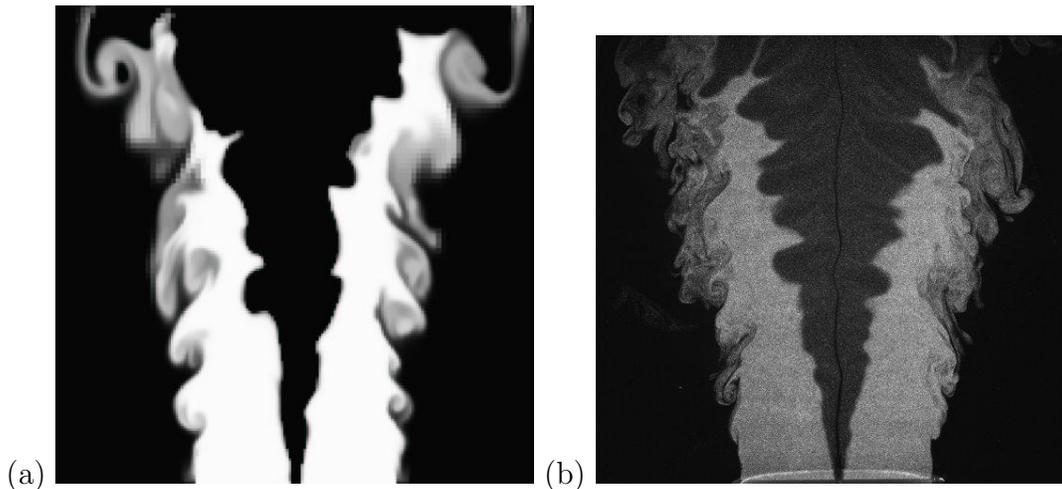


Figure 3: (a) *Computed mole fraction of  $CH_4$* , (b) *Photograph of illuminated particles seeded in fuel stream*

We are continuing our comparison with experimental data to provide a more detailed comparison with simulation and experiment and to identify the key factors controlling the turbulent flame speed in this flow regime [7].

### 2.2.3 Future Plans

Our short term focus in combustion is to complete the analysis of turbulent V-flames. We are in the process of completing simulations for a variety of inflow turbulence configurations. In the first stage of the analysis we will focus on validation with the experimental data, flame morphology and flame dynamics. Subsequent analyses will examine the details of the turbulence / chemistry interaction.

The next problem we plan to simulate is a turbulent “low-swirl” premixed flame. In the low-swirl configuration, air is injected at high speeds tangential to the interior walls of the nozzle. Above the nozzle exit, the expanding fuel-air mixture create an axial velocity deficit in the central core capable of stabilizing a premixed flame. Low-swirl nozzles have considerable practical value because the resulting flame has very low pollutant emissions. The experimental configuration is similar to that of the V-flame; however the interactions between the fluid dynamics and the volumetric expansion of the flame are considerably more delicate and difficult to simulate. Our initial goal for will be to compute a statistically stable flame and analyze the basic flame dynamics.

Our previous work on modeling laboratory-scale premixed turbulent combustion has been based on a description of chemical kinetics given by DRM-19 which contains 21 species and 84 reactions. This mechanism provides a good approximation to the basic carbon pathways in the flame; however, it includes some simplifications for the carbon chemistry and does not include any of the nitrogen chemistry needed to model pollutant formation. Our longer term plans are to revisit the two experimental flames we have considered and introduce a more complete treatment of chemical kinetics, including the nitrogen chemistry needed to model formation of  $NO$  and  $NO_2$  in methane flames.

In addition to these computational studies our other long term objective is to incorporate realistic geometries into our reacting flow simulation methodology. Our goal is to be able to represent the full geometry of laboratory combustion experiments such as nozzles, flow obstructions and confining containers. This capability will allow us to perform detailed simulations of experiments without compromises in fidelity resulting from simplifying assumptions about the flow domain. For this development we will utilize geometric capability being developed in the center which will have reached a suitable level of maturity to be incorporated into our reacting flow methodology.

## 2.3 Magnetic Fusion

### 2.3.1 Goals and Approach

In this section we describe the development of an adaptive mesh refinement (AMR) MHD code for fusion applications. The Chombo framework developed at LBNL is used for AMR. The MHD code solves the equations of hydrodynamics and resistive Maxwell’s equations written below in conservation form:

$$\frac{\partial U}{\partial t} + \frac{\partial F_j(U)}{\partial x_j} = \frac{\partial F_{v,j}(U)}{\partial x_j},$$

where the solution vector  $U \equiv U(x, y, t)$  is,

$$U = \{\rho, \rho u_i, B_i, e\}^T,$$

and the flux vectors  $F_j(U)$  and  $F_{v,j}(U)$  are given by

$$F_j(U) = \left\{ \rho u_j, \rho u_i u_j + \left(p + \frac{1}{2} B_k B_k\right) \delta_{ij} - B_i B_j, u_j B_i - u_i B_j, \right. \\ \left. (e + p + \frac{1}{2} B_k B_k) u_i - B_i (B_k u_k) \right\}^T,$$

and

$$F_{v,j}(U) = \left\{ 0, \tau_{ij}, \eta \frac{\partial B_i}{\partial x_j} - \eta \frac{\partial B_j}{\partial x_i}, \tau_{ij} u_i + \kappa \frac{\partial T}{\partial x_j} + \eta \left( \frac{1}{2} \frac{\partial B_k B_k}{\partial x_j} - B_i \frac{\partial B_j}{\partial x_i} \right) \right\}^T.$$

In the above equations,  $\rho$  is the density,  $u_i$  is the velocity,  $B_i$  is the magnetic field,  $p$  and  $T$  are the pressure and temperature respectively, and  $e$  is the total energy per unit volume of the plasma. The equations are closed by the following equation of state

$$\epsilon = \frac{p}{\gamma - 1} + \frac{\rho}{2} u_k u_k + \frac{1}{2} B_k B_k.$$

The stress tensor is related to the strain as

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}.$$

The plasma properties are the viscosity  $\mu$ , the conductivity  $\kappa$ , and the resistivity  $\eta$ .

The hyperbolic portion of the above equations are solved numerically using a combination of the 8-wave upwinding method [35] and the unsplit numerical method [10]. The solenoidal property of the magnetic field is enforced by means of a projection method. A salient feature of the present method is that the projection of the magnetic field is not done in the spirit of “divergence-cleaning” but that it is intimately coupled with the construction of the hyperbolic fluxes. The diffusive (or parabolic) portion of the above equations are solved either explicitly or implicitly. The implicit implementation results in variable coefficient Helmholtz equations which are solved using a multi-grid method. By using the appropriate solver components, we can easily switch between backward Euler, Crank-Nicholson or implicit Runge-Kutta [37] in the implicit treatment of the parabolic fluxes. We have endeavoured to be faithful to the strong conservation form in the numerical method. Details concerning the numerical method will be discussed in an upcoming publication.

The AMR MHD code was applied to the applications discussed below. In both applications, the AMR computations afforded a resolution which would have been quite computationally expensive or prohibitive if mesh adaptivity was not used. Speed-up of computations due to AMR were estimated to range from a factor of 6 to 40.

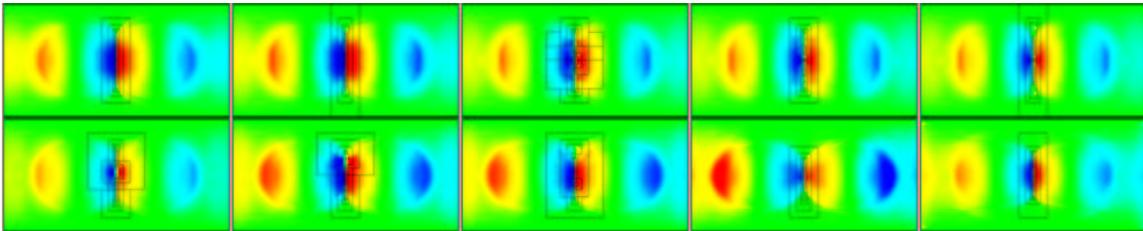


Figure 4: Time sequence of the y-component of the magnetic field. The black bounding boxes depict the meshes in the AMR hierarchy.

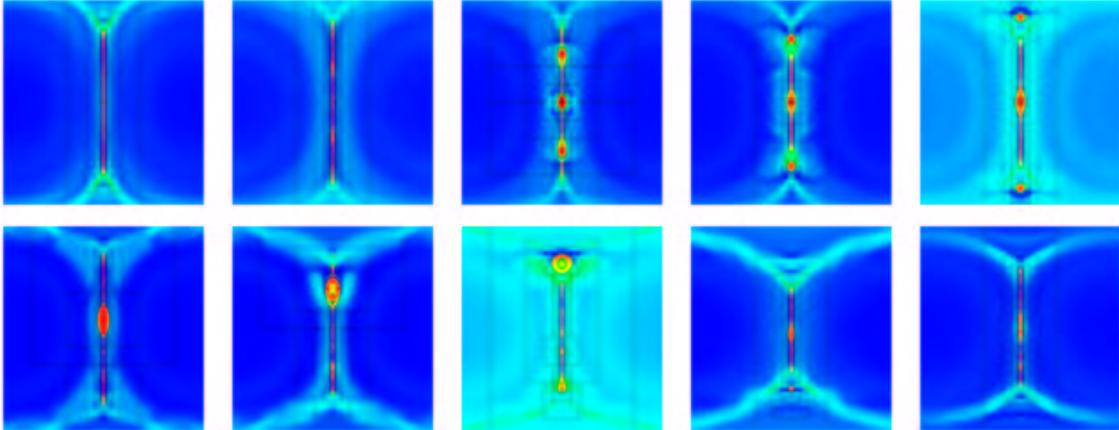


Figure 5: Time sequence of the perpendicular current corresponding the previous figure. The current layer is unstable which causes ejections of high pressure plasma from the reconnection layer.

### 2.3.2 Accomplishments

#### Magnetic Reconnection in 2D.

Magnetic reconnection (MR) refers to the breaking and reconnecting of oppositely directed magnetic field lines in a plasma. In the process, magnetic field energy is converted to plasma kinetic and thermal energy. MR occurs in many contexts: for example, in the sawtooth-like oscillations observed in the operation of a tokamak, and in solar coronal events. In general, in magnetic reconnection, two regions are distinguished: an outer “inviscid” region and an inner “resistive” region, whose width scales with  $\eta^{\frac{1}{2}}$ , where the actual breaking and reconnecting of the magnetic field lines takes place. We have carried out simulations of magnetic reconnection in an idealized canonical two dimensional setting. The objective is to solve the full single-fluid resistive MHD equations and consider both the inner and outer regions. AMR is an obvious computational methodology to resolve the near-singular current sheets.

We now discuss some recent observations in AMR simulations of reconnection at Lundquist number  $S = 10^4$ . A time sequence of the y-component of the magnetic field is shown in Figure 4. As the two flux tubes move in closer we observe an intensification of the perpendicular current. During reconnection, the current layer becomes unstable and results in the ejection of high pressure plasma with the ejection direction alternating between the top and bottom (See Figure 5). The peak reconnection rate during this phenomenon is larger than the theoretical reconnection rates. It is still an open question whether such a mechanism is partially responsible for observed reconnection rates which are faster than those suggested by the Sweet-Parker scaling.

#### Magneto-hydrodynamic Richtmyer-Meshkov Instability.

The Richtmyer-Meshkov (RM) instability is the subject of extensive experimental, theoretical and computational research due to its importance in technological applications such as inertial confinement fusion, as well as astrophysical phenomena such as supernovae collapse. The RM instability is one of the main inhibiting mechanisms in inertial confinement fusion. We performed several AMR simulations of the Richtmyer-Meshkov instability, both

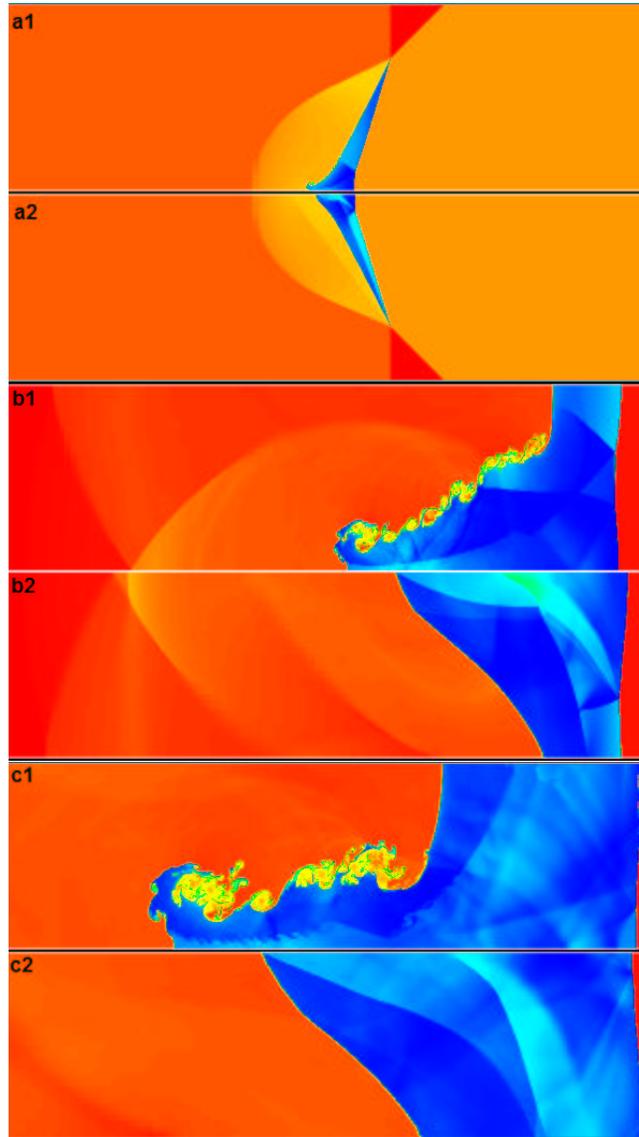


Figure 6: Time sequence of the density field in the Richtmyer-Meshkov instability at times  $t = 0.385$  (a1,a2),  $t = 1.82$  (b1),  $t = 1.86$  (b2),  $t = 3.33$  (c1), and  $t = 3.37$  (c2). The images with nonzero magnetic field (a2, b2, c2) are reflected about the x-axis.

in the presence and absence of a magnetic field. The mathematical model is that of the ideal MHD. A time sequence of the density field is shown in Figure 6 for the non-magnetic RM instability (Figure 6:a1,b1,c1), as well as the evolution in the presence of a magnetic field (Figure 6:a2,b2,c2). The top two images depict the early refraction process of the incident shock at the contact discontinuity during which the interface is compressed by the incident shock and baroclinic vorticity generation takes place. The middle two images (Figure 6:b1,b2) show the development of the instability at a later time  $t \approx 1.8$ . In the absence of the magnetic field the interface, which is a vortex layer, rolls-up as expected for the usual Richtmyer-Meshkov instability. In the presence of the magnetic field, the interface remains smooth and no evidence of roll-up is observed. The bottom two images show that the interface, in the absence of a magnetic field, has grown in extent and shows a considerable amount of mixing. On the other hand, in the presence of the magnetic field, the average extent of the interface shows no difference between this time and the earlier one. The physical explanation of the suppression is that in the presence of a magnetic field, the vorticity generated at the interface is carried away by a pair of slow magnetosonic shocks. Consequently the interface is devoid of vorticity, and its growth and associated mixing is completely suppressed.

### 2.3.3 Future Plans

Pellet injection in tokamaks was identified as the most suitable application for the AMR MHD code. AMR is essential to provide the resolution required to simulate realistic pellet sizes relative to device dimensions (typical ratios are  $O(10^{-3})$ ). Experimentally, it is known that the density distribution, after the pellet ablates upon encountering the high temperatures in a tokamak, is not consistent with the distribution inferred from assuming that the ablated material remains on flux surfaces where ablation occurred. The subsequent redistribution of mass is due to MHD processes. Our goal is to characterize these processes.

In the longer term, there are a number of other extensions to our current capability that would be desirable.

- Continue development of the semi-implicit AMR MHD solver to include nonlinear resistivity. Perform magnetic reconnection runs at higher Lundquist numbers.
- Include a model for pellet ablation and perform high resolution pellet injection in Cartesian geometry as well as with models to mimic toroidal effects.
- Currently the hyperbolic terms in the MHD equations are treated explicitly. We will investigate a fully implicit formulation or implicit treatment of the fast waves.
- Investigate methods to include the Hall term in the MHD equations.

## 2.4 Accelerator Modeling

### 2.4.1 Goals and Approach

As discussed in section 3.1.1 of the original proposal, the goal in this project is to develop two different simulation capabilities related to accelerator design. For Vlasov-Poisson equations for beam dynamics, the goal is to develop a version of the particle-in-cell method in which the Poisson equation for the electrostatic field is solved on an AMR grid. The potential

speedup of such a code is substantial. The grid resolution is determined by the requirement the number of particles per grid point is bounded from above (around 10), while the size of the domain is determined by other factors, e.g. boundary conditions. In practice, only a small fraction of the field domain contains particles, and the remainder could be adequately resolved with far fewer grid points without any loss of accuracy in the particle trajectories.

The second goal is to develop a simulation capability for transient gas jet problems arising in the simulation of laser-driven plasma-wakefield accelerators. This problem is a fluid dynamics problem, requiring a solution to the unsteady compressible flow equations in complex geometries. These simulations will use the embedded boundary software described in section 3.3 of this document.

### 2.4.2 Accomplishments

We have developed a prototype implementation of an AMR-PIC method (ChomboPIC) (figure 7). There are several components to this method. One is the node-centered AMR elliptic solver described in section 3.2 of this document. A second is a generalization of algorithms for depositing charge on the mesh and of interpolating electric field of the particles on a uniform grid to the case of an AMR grid. The third is the use of the Chombo Layer 1 data structures for distributed bin-sorted particles on AMR grids. We have coupled this prototype to the MaryLie / IMPACT beam dynamics code developed by Ryne and others as part of the SciDAC Accelerator Modeling project. Other initial target applications of ChomboPIC are the WARP and MAD9 beam codes. We have begun to compare in detail the results obtained from the original MLI/PIC code to those obtained by replacing the uniform grid Poisson PIC solver with ChomboPIC. The principal difference between the two codes is the Poisson solver: ChomboPIC uses a finite-difference algorithm versus the spectrally accurate FFT convolution algorithm used by MLI. We are now in the process of performing a careful and systematic comparison between the two codes, to obtain quantitative estimates of the impact of the change in discretizations on the overall accuracy of the method in real beam dynamics problems.

In the area of gas-jet simulations for plasma-wakefield accelerators, the embedded boundary codes are just reaching a point where we will be able to begin simulations of these problems, and compare them to experiments. We have been working with Eric Esarey to evaluate our grid generation tools. In figure 8, we show the results of taking a geometry specification for a nozzle geometry, and applying the surface generation and Cart3D grid generation tools.

### 2.4.3 Future Plans

In the area of beam dynamics, future efforts will focus on two areas. The first will be to investigate the effect on the long-time particle trajectories of the differences in the electric field computed by the different methods. To do this, we will implement a fast version of James' algorithm for computing infinite domain boundary conditions [23] that we will use for both codes. This will, in addition, speed up the MLI uniform grid Poisson solver by a factor of 5 or more. At that point we should have sufficient information to know under what circumstances AMR-PIC algorithms will be worthwhile substitutes for FFT-based ones.

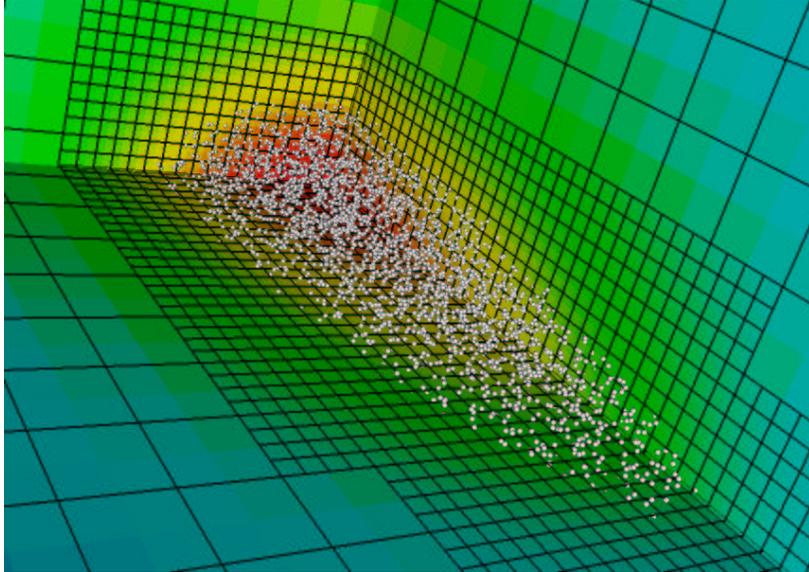


Figure 7: AMR-PIC calculation. The electrostatic potential induced by the particles is displayed on the slicing planes.

Over the next year, we expect to be able to provide a complete suite of simulation capabilities for the gas-jet problem for plasma-wakefield accelerators in 3D and 2D axisymmetric geometries, with both inviscid and viscous capabilities.

## 3 Software Development

### 3.1 Overview

One of the principal characteristics of the algorithms being developed here is that they are difficult to implement: they are more complicated than traditional finite difference methods, and the data structures involved are not easily represented in the traditional procedural programming environments used in scientific computing. To manage this algorithmic complexity, we use a collection of libraries written in a mixture of Fortran and C++ [16] that implements a domain-specific set of abstractions [22] for the combination of algorithms described above. In this approach, the high-level data abstractions are implemented in C++, while the bulk of the floating point work is performed on rectangular arrays by Fortran routines.

The starting point for the software development part of APDEC is the Chombo software library, developed by the Applied Numerical Algorithms Group at LBNL as part of the Berkeley Lab AMR release. This library is a refactored version of the BoxLib family of libraries, developed by the Center for Computational Sciences and Engineering at LBNL. Chombo has been specifically designed to make it easier to extend AMR to support embedded boundaries and particle methods, as well as adding new capabilities and new implementations of AMR algorithms on rectangular domains.

We would like to comment on the management of the software development process. By

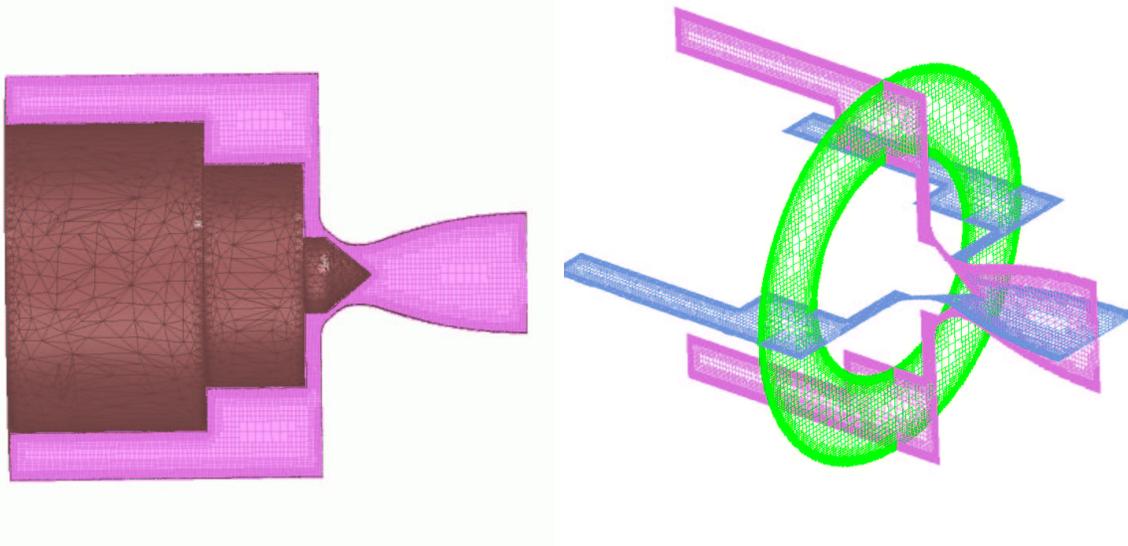


Figure 8: Embedded boundary grid generation for gas jet nozzle. The left image shows the surface triangulation obtained from RAP, and the right image the cut-cell generation from Cart3D.

commercial software engineering standards, the software development effort described here is at most of medium size: a few hundred thousand lines of code, and with fewer than ten developers. However, by scientific computing standards, this is a project of extraordinary complexity. High performance (both serial and parallel) is difficult to obtain. The algorithm space is complex, and has a high degree of uncertainty. We are combining these components in new ways for these applications. In addition, some of the components such as the embedded boundary solvers are themselves new. In this environment, we have found it necessary to impose a more disciplined software development process than has been customary in the scientific computing community. Our current practice conforms to that discussed in section 6.1 of the original proposal. In addition, we have maintained a practice of writing design documents for algorithms and APIs for major components prior to beginning implementation, and to keeping those documents updated. These documents are a critical component of our software engineering process, providing the mechanism by which the developers communicate with one another. They are referred to throughout the present document, and are accessible at the project web site.

## 3.2 AMR Software for rectangular-grid computations

### 3.2.1 Goals and Approach

The goal of this part of the project is to develop a software framework for block-structured adaptive mesh refinement algorithms for partial differential equations in rectangular domains.

Our approach to the design of the Chombo library is that of a layered architecture, which is described in section 3.3 of the original proposal. However, there are specific design features that have proved particularly useful in the development of the framework that we would like to point out here. We shall give only a brief overview; a more detailed discussion can be

found in the Chombo design document [11].

#### Classes to represent indexing constructs.

Much of the programming complexity of representing irregular stencil operations arising in AMR can be expressed as iterations over irregular sets of indices that can be computed using set calculus on  $\mathbb{Z}^D$ , where  $D$  is the spatial dimension. Chombo provides a set calculus for rectangular index sets (`Box` class) and irregular index sets (`IntVectSet` class). These provide a programming notation that is close to the mathematics with which one describes the algorithms.

#### SPMD parallelism with a decentralized representation of communication.

Multidimensional arrays in Chombo consist of a `Box` describing the rectangular index set in space, plus a contiguous block of data to store the array values. That pattern is repeated in the description of data defined on a union of rectangles distributed over processors. The distribution of rectangles onto processors is contained in objects in the `BoxLayout`, `DisjointBoxLayout` classes, while the data objects (`BoxLayoutData`, `LevelData` classes) are defined as a collection of multidimensional arrays distributed over processors as specified by the `BoxLayout`. The only distributed communications operations are ones that copy on intersection from a `LevelData` to a `BoxLayoutData`. Otherwise, access is restricted to the data on ones own processor, and is managed through iterators. Rather than having a global repository of which distributed objects have isomorphic processor distributions, this information is inferred at runtime: data objects can be accessed with the same iterator if the `BoxLayouts` are the same, or if they are both derived from a common layout by transformations such as coarsening and refinement. Such a decentralized representation of distributed data make for a flexible and familiar programming idiom. One can create temporaries and maintain disciplined scoping rules in much the same way one would in a serial program, and still obtain reasonable performance.

#### Library support for coupling between refinement levels.

Some of the most complicated code in AMR is that which manages the coupling between levels in a grid hierarchy, such as averaging and interpolating between levels of refinement, or interpolating boundary conditions at boundaries between different levels of refinement. This is because such operations combine irregular computation and parallel communication. Fortunately, such operations are reuseable across applications. For example, the algorithms and data structures used to maintain conservation at interfaces between refinement levels is the same, whether the problem is hyperbolic or elliptic. For that reason, we have separated out such operations as a distinct layer (Layer 2) in the Chombo design.

#### Templated data containers and interface classes as mechanisms for code reuse.

We make extensive use of C++ templates in order to obtain code reuse. For example, `LevelData<T>` is templated over the class that is the analogue of a multidimensional array. The class `T` must provide only a small number of member functions that form the interface to MPI used in `LevelData`. This allows us to reuse the code that manages distributed data over a broad range of data types, e.g., regular arrays with various centerings, embedded boundary data, or binned particles. Similarly, we use interface classes, a standard abstraction in object-oriented design, to obtain reuse of complicated control structures. A good example of this is the `AMR / AMRLevel` pair of classes that manage the Berger-Oliger timestepping

algorithm for AMR with refinement in time. The `AMR` class implements the timestepping algorithm, calling member functions of a vector of pointers (one for each level of refinement) to `AMRLevel` objects. In that way, any class that implements the `AMRLevel` interface, such as advancing the solution in time, tagging cells for refinement, can then use Berger-Oliger timestepping. This pair of classes is used extensively in the Berkeley Lab AMR codes, in applications ranging from incompressible flow and low Mach number combustion to shock waves in solids.

One of the important consequences of using the Chombo framework for implementing AMR applications is that codes tend to be “born parallel”. The `LevelData` class for representing data on a union of rectangles imposes a disciplined access to off-processor memory, and the Layer 2 infrastructure hides the parallel communication for interlevel operations. If an application conforms to these APIs, little additional work is required to make it into a correct parallel program.

### 3.2.2 Accomplishments

Over the last 18 months, the focus has been on three specific aspects of AMR software development. The first has been to provide the components required to implement the AMR MHD and AMR-PIC methods in support of our stakeholders in magnetic fusion and accelerator modeling, including hyperbolic and elliptic solvers, support for particles, and various tools required to support such calculations, e.g. visualization tools. The second has been to address some of the major interoperability issues of making the software that we have developed more widely accessible and of coupling the tools being developed in the other SciDAC ISICs and elsewhere to our framework. The third has been to design of the rectangular domain AMR tools in a way so as to maximize their reuse in the embedded boundary geometry versions of the algorithms discussed below, and more generally to use the design process for the rectangular domain tools to resolve some of the design issues for the complex geometry case.

#### Solver packages.

We developed a variety of iterative solvers in Chombo for cell-centered AMR discretizations of elliptic and parabolic PDE. These included solvers for variable coefficient elliptic equations, as well as a tensor solver for implicit discretization of viscous terms in compressible flows. We also implemented several implicit Runge-Kutta time discretization packages for parabolic equations on unions of rectangles, including backward Euler, Crank-Nicolson, and the second-order accurate  $L_0$ -stable method of Twizell, Gumel, and Arigu [37].

We developed a general package for the unsplit higher-order Godunov methods for hyperbolic conservation laws described in [10, 36]. These methods have a number of properties that make them attractive for the applications described here, such as low phase error for accurate treatment of time-dependent wave propagation, robust treatment of shocks and underresolved gradients, and a straightforward construction of stable coarse-fine boundary conditions for AMR. In addition, these methods have straightforward extensions to semi-implicit treatment of diffusive terms [5, 27]. The package developed here [15] requires the user to provide a small amount of problem-dependent code (characteristic analysis package

and Riemann solver). The rest of the discretization is independent of the application, and hence handled by the package. This package as well as the elliptic and parabolic packages described above, are being used in MHD applications described in the applications section of this document. We also used the design of this package as a starting point for the design of the corresponding package for the embedded boundary case.

For particle-in-cell methods, nodal point centering is natural, and the customary usage: the bins into which the particles are sorted are cell-centered, and the charges and fields on the grid are computed at the nodes adjacent to the bins. For that reason, we implemented an AMR multigrid package for solving Poisson’s equation with nodal-point centering [26]. Capabilities include combinations of Dirichlet and periodic boundary conditions in the various coordinate directions, and a treatment of embedded irregular Dirichlet boundaries based on a Shortley-Weller treatment of the irregular boundary conditions (figure 9). This is the principal solver component for the AMR-PIC package for Vlasov-Poisson for accelerator modeling.

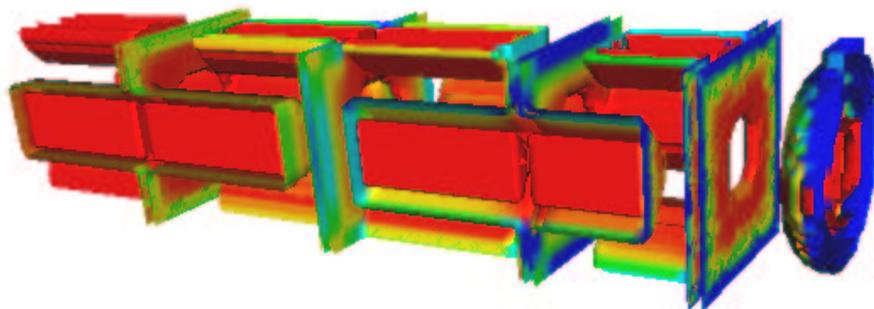


Figure 9: Nodal-point solution of Poisson’s equation with Shortley-Weller treatment of geometry. The conducting surfaces of a high-current ion-beam accelerator are colored with the electric field strength.

### Infrastructure Support

There were a number of Layer 1, Layer 2, and Utility Layer capabilities that we developed, primarily in response to the needs of various applications or of other software infrastructure developers. We developed Layer 1 tools for particles, including distributed arrays of binsorted particles defined on unions of rectangles, as well as HDF5 I/O for particles. We also extended the Chombo framework to support periodic boundary conditions in one or more directions. We did this by incorporating the notion of periodicity into our fundamental definitions of index spaces. Once that was done, it was straightforward to extend the the Chombo infrastructure, up through solvers and complete applications, to support periodic boundary conditions.

We have developed an extensive suite of visualization tools specifically designed for displaying block-structured data, called ChomboVis, primarily under funding from NASA. However, there are a number of capabilities that were developed specifically in response

to the needs of APDEC. These include particle visualization, and the ability to invoke the visualization tool dynamically from the debugger.

#### Interoperability Issues

We developed Chombo API's for the PERC ISIC's PAPI tools for accessing hardware performance counters, and of a version of the Chombo solvers for elliptic PDE on a single level of refinement for that uses the *hypre* package from the TOPS ISIC as its the core iterative solver. We have also incorporated the University of Washington CLAWPACK package for hyperbolic conservation laws into the Chombo AMR infrastructure. CLAWPACK is a highly successful package in the hyperbolic PDE community, with an accessible API for such problems. This will facilitate comparison of the different solvers, and will also aid the user community by making a wider set of tools available. Researchers who are already using CLAWPACK should be able to easily switch to Chombo to take advantage of features that CLAWPACK lacks, such as parallel processing of adaptive grids and the adaptive elliptic and parabolic solvers available in Chombo.

We have been working with the larger block-structured AMR community to develop tools that would allow us to more easily use software packages developed using different frameworks. In April, 2002, representatives from a number of the large AMR development groups met under the auspices of the CCA Forum. The outcome of that meeting was a proposed data alias based on opaque handles that allows one to pass pointers to AMR data in a framework-neutral fashion. We have since implemented that data alias, and have used it to wrap the Chombo interface to HDF5 I/O. The latter was at the request of several groups who want to use the ChomboVis visualization tools.

### **3.2.3 Future Plans**

#### Design and Performance Review

Our first priority after the present review will be to undertake a complete design and performance review of the Chombo framework for rectangular domain problems, using the experience of our applications developers as a driver. There are a number of soft spots that we can already see. In the performance area, we expect to undertake a reimplementaion of some of the Layer 2 software for serial optimization of irregular codimension-one stencil operations, and to develop a variety of AMR grid-generation tools to optimize load balancing. We also need to revisit the design of the unsplit hyperbolic package to allow it to accomodate more easily the coupling of elliptic and hyperbolic steps, as arises, for example, in the MHD algorithm. We expect to collaborate with investigators in the PERC ISIC in carrying out this work.

#### Interoperability Tools

We will begin a collaboration with the CCTTSS ISIC to wrap the AMR elliptic solvers as CCA components. We will continue our efforts to couple Chombo to CLAWPACK, extending the interface to support non-conservative quasilinear hyperbolic equations and the algorithms for mapped grids currently under development. We will work with the *hypre* team under TOPS to extend the *hypre* semistructured API to support multigrid solvers on AMR hierarchies, and to improve the performance of the *hypre* structured solvers on unions of rectangles typical of those arising in AMR applications.

### Solvers.

We will begin development of elliptic and parabolic AMR solvers for problems involving coefficient anisotropies aligned with the coordinate directions. Such problems arise in magnetic fusion, in which diffusive transport enhanced in the direction of the magnetic field, which is approximately aligned with the toroidal direction. Other applications include accelerator modeling, due to the anisotropy of the beam, and geophysical fluid dynamics, where the coefficient anisotropies arise from the large aspect ratio between the horizontal and vertical scales.

We will begin development of parallel AMR software for constant-coefficient elliptic equations, based on the method of local corrections approach described in [3]. In this approach, one uses potential theory to construct a domain decomposition method that requires only one iteration between the local solves and the global coarse solve. This is comparison to the more traditional iterative approaches which require many iterations between the local and global solutions. We expect this to be an essential component to scaling up the AMR-PIC codes.

## **3.3 Cartesian Grid Embedded Boundary Methods and Software for PDEs in Irregular Geometries**

### **3.3.1 Goals and Approach**

The goal of this part of the project is to extend the framework described above to the case of embedded boundary methods for problems with irregular geometries. The underlying algorithmic approach is described briefly in section 3.2.2 of the original proposal.

One approach to extending the framework described above to the case of embedded boundaries would be to store all solution data in rectangular arrays using the framework described above, under the assumption that there is only one control volume per rectangular grid cell. There are two difficulties with such an approach. The most serious one is that it unacceptably restricts the geometries that can be represented. Thin bodies - ones whose width is less than the mesh spacing - can divide a grid cell into two control volumes, as can a boundary with a sharp corner. Furthermore, even if one is willing to accept the restriction on the geometry to eliminate those cases at the finest grid level, they typically reappear as one coarsens the mesh (as is done for both AMR and multigrid). In addition, it is often necessary to define variables on sparse subsets of the domain, e.g. the irregular control volumes only, leading to substantial memory overheads.

For this reason, we have taken an alternative approach that accomodates at the outset the possibility of multiple control volumes per rectangular grid cell [14, 13]. It is based on a set of abstractions for Cartesian grid embedded boundary methods that generalizes the mathematical structure of rectangular grid calculations implicit in the design of Chombo. On ordinary rectangular grids, the fundamental object used to define both variables and operations are multiple copies of  $\mathbb{Z}^D$ , related to one another in a nested hierarchy by coarsening and refinement by some fixed factor (in our case, that factor is 2). All of the rest of the machinery of Chombo is derived from the fundamental properties of this index space hierarchy: set calculus, arrays defined over rectangular subsets, `BoxLayout`, `LevelData`, etc. For the case of embedded boundaries, we replace the hierarchy of copies of  $\mathbb{Z}^D$  by a hierarchy

of graphs in which the nodes of the graph are the control volumes, and the arcs of the graph are the faces connecting control volumes in adjacent cells. In addition, the graph has a mapping into  $\mathbb{Z}^D$  that defines which rectangular grid cells and faces contain the corresponding control volumes and faces of the graph. This mapping allows us to define subsets of the graph corresponding to both rectangles and arbitrary irregular subsets of  $\mathbb{Z}^D$ . From there, we define rectangular arrays as maps from the appropriate subset of the nodes or faces of to some value space; we also provide arrays defined over arbitrary subsets of  $\mathbb{Z}^D$ . These arrays are indexed by node or face index objects that generalize the tuples of integers used to index ordinary arrays. Such arrays also conform to the requirements of the template type in `Leveldata<T>`, so that we can define distributed data objects over unions of rectangles.

This design is sufficiently general to represent multiple control volumes per cell, while still providing an expressive notation for indexing into such structures. The graph representation also simplifies grid generation to some extent. The topological and geometric information required to compute various discretized operators is generated at the finest level, where the geometry is most completely resolved on the grid, from a CAD description of the surface or from a local functional description. At coarser levels, this information is generated by querying the graph representation at the next finer level, without having to access the original geometric description of the surface. The array types are implemented as ordinary rectangular arrays that are accessible to the user, augmented by a sparse irregular data structure containing the values for cells with multiple control volumes. This enables one to call Fortran routines on rectangular grids to obtain high-performance calculations for operations on regular cells, with the sparse irregular calculations implemented in C++. Finally, the distributed version of data on unions of rectangles is obtained with the minimal effort required to implement the API's required by `LevelData`.

### 3.3.2 Accomplishments and Status

Over the first 18 months of this project, the effort in this area has been divided between extending the low-level layers of the EBChombo infrastructure, developing grid generation tools, and producing initial implementations the embedded boundary versions of the core AMR algorithms for hyperbolic and elliptic PDE.

#### EBChombo Infrastructure.

We designed and implemented the Layer 1 and Layer 2 infrastructure for the embedded boundaries, following the approach described above. For Layer 1, this included classes for managing the graph representation of the embedded boundary index space; cell-centered and face-centered array types defined over both rectangles and irregular sets of indices; and distributed-memory aggregates of such arrays, using the `LevelData` API. The Layer 2 infrastructure includes classes for managing interlevel operations analogous to those in Chombo, such as averaging and interpolation between levels, and the various coarse-fine boundary condition operators required to implement multilevel elliptic and hyperbolic operators. In addition, we implemented other operators specific to embedded boundary applications, such as multilevel redistribution operators for maintaining stability and conservation between levels in hyperbolic problems.

#### Embedded boundary solvers.

In the hyperbolic case, we developed a extension of the second-order Godunov method for

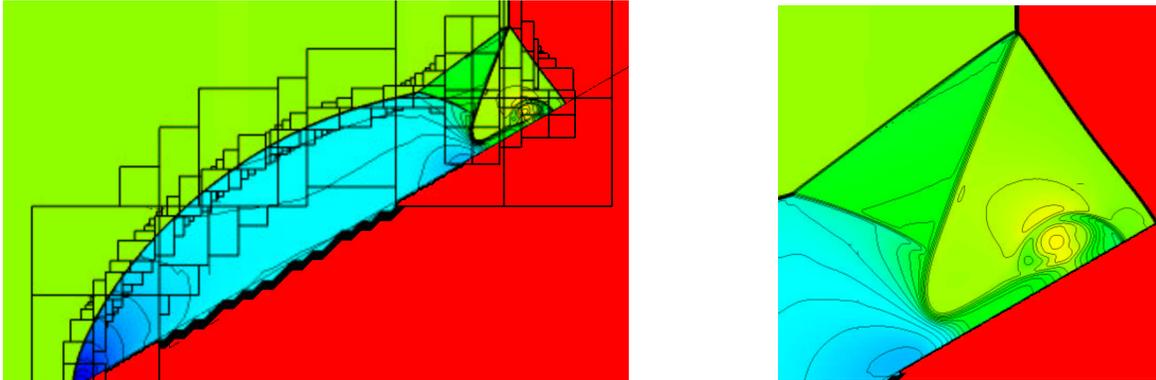


Figure 10: Shock-reflection problem computed using the EBAMR Godunov code. The oblique reflecting surface is represented using the embedded boundary method on a rectangular grid. Left: full density field, with grid outlines superimposed. Right: close-up of double-Mach region.

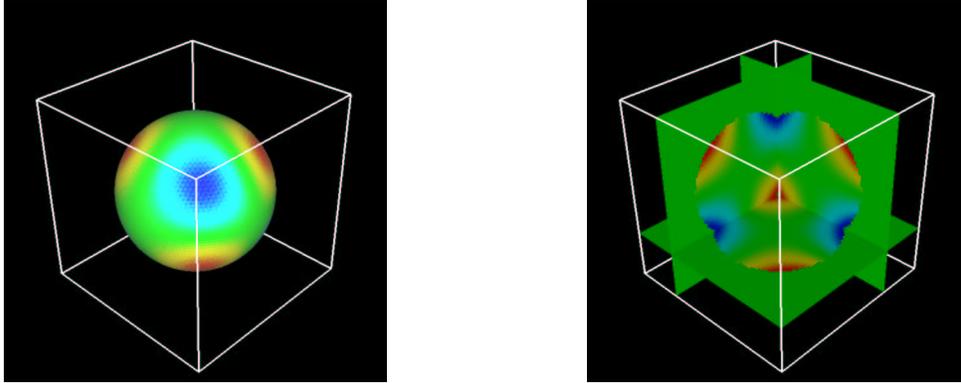
discretizing hyperbolic conservation laws to the case of embedded boundaries that is formally consistent, i.e. for smooth solutions, the truncation error vanishes with the mesh spacing as the mesh spacing approaches zero at the embedded boundary [12]. When combined with a second-order accurate method in the interior, we obtain a method that has a second-order accurate solution error in  $L^1$ . The method is also robust in the presence of strong discontinuities. We have also developed an adaptive mesh refinement version of this method, using the Layer 1 and Layer 2 tools described above (figure 10). We are currently developing 3D and 2D axisymmetric versions of this algorithm as design tools for laser-driven plasma-wakefield accelerators.

In the case of elliptic equations, we developed an extension of the discretization in [24] to three dimensions based on the use of bilinear interpolation to obtain second-order accurate fluxes at the centroids of faces. We have implemented and tested this algorithm for the case of a single level grid, and have found that it is stable and leads to a second-order accurate solution error (figure 11). We are in the process of extending this algorithm to the adaptive case. As part of the work on the adaptive version, we are refactoring the design of the overall Chombo AMR multigrid solver framework, to make iterative solver control structures reusable across the type of the data being solved for. This will, in particular, allow us to use the same implementation for both rectangular domain and embedded boundary problems.

#### Grid Generation.

We began development of an integrated capability to produce embedded boundary (EB) mesh geometry from general geometric descriptions. The previously available technology required the user to provide a description of the geometry in terms of a water-tight surface triangulation for all embedded objects. Using technology derived from the Rapsodi project at LLNL, we enhanced this capability so that EB mesh geometry can also be constructed from CAD IGES files, or by using an interactive tool for constructing geometries from simple shapes. Using Rapsodi's geometry representation also opens up the opportunity for generating higher-order EB surface information, and for dynamic generation of EB geometry during an AMR-based calculation.

The Cart3D package from NASA-Ames [2] is used to construct embedded boundary ge-



grid	$\ \epsilon\ _\infty$	$p_\infty$	$\ \epsilon\ _2$	$p_2$	$\ \epsilon\ _1$	$p_1$
$16^3$	$4.80 \times 10^{-4}$	—	$5.17 \times 10^{-5}$	—	$1.83 \times 10^{-5}$	—
$32^3$	$1.06 \times 10^{-4}$	2.17	$1.25 \times 10^{-5}$	2.05	$4.41 \times 10^{-6}$	2.05
$64^3$	$2.43 \times 10^{-5}$	2.13	$3.07 \times 10^{-6}$	2.02	$1.09 \times 10^{-6}$	2.02

Figure 11: Solution of Poisson’s equation in a sphere using embedded boundary methods. The solution error on the surface of the sphere is shown in the left image, and the solution error in slices through the sphere in the right image. The table contains convergence data in  $L^1$ ,  $L^2$ , and max norms.

ometry information from water-tight surface triangulations. The RAP software from the Rapsodi project [33] was modified so that surface triangulations appropriate for input into Cart3D are constructed. Following the initial implementation, a new more robust topology algorithm was implemented for building water-tight surface triangulations. RAP allows the user to interactively process CAD IGES files to remove errors and de-feature the geometry in preparation for simulation and analysis. [34] This now includes a capability to automatically identify and repair common geometry defects. Algorithms were developed and implemented to automatically treat singular surface patches in IGES geometry representations, further increasing the robustness of the software. RAP also provides tools for interactively constructing simple geometries, avoiding the use of complete CAD packages for cases where that level of sophistication is not needed.

At the initiation of this project, RAP was built upon the Overture Framework, a large software framework that supports the development of PDE solvers based on the method of overset grids, and similar largely-structured meshing approaches. In order to provide a more compact implementation for the APDEC project, the geometry functionality in the Overture Framework was decoupled into a smaller library, now called Rapsodi. The new version of RAP, including all of the features mentioned above, was released publicly on the Web as part of Overture v. 19.

### 3.3.3 Future Plans

#### Embedded Boundary Solvers and Software.

We will complete the development of the EB AMR elliptic solvers. We will combine them with the hyperbolic solvers to produce a compressible Navier-Stokes solver for the gas-jet

application, and begin validation of the code in collaboration with the scientists working in that area. At that point, we will have been through one development cycle of the embedded boundary method, and it will be appropriate to undertake a design and performance review similar to that planned for the rectangular domain case. We expect similar issues to arise here, such as improving the serial performance of sparse irregular calculations, and developing new load-balancing techniques to accommodate the more irregular loads introduced by the calculations at the embedded boundary.

There are a number of directions in which we will need to extend the embedded boundary approach, in order to meet the needs of our applications stakeholders. We will generalize the capabilities in the embedded boundary grid methods, to include the anisotropic grids and general orthogonal coordinate systems. Both capabilities are needed to apply embedded boundary methods to tokamak and stellarator geometries, and a 2D cylindrical coordinate capability would be a useful tool for the gas jet problem. To support combustion, we will augment our solvers to accommodate the complicated interactions of hyperbolic transport and elliptic constraints that arise in low-Mach number models. We will begin to generalize the EB AMR infrastructure to support volume-of-fluid representations of free boundaries, i.e., ones for which PDEs are being solved on both sides of an irregular surface. Such methods would be useful for computing thin tracked flames, and for representing the boundary of the gas jet propagating into a vacuum.

#### Grid generation.

The current grid generation process involves a number of steps using a number of different codes. First the geometry is read from a CAD IGES file or created internally in the program RAP. That program makes sure the surface description represents a water tight model and generates a triangulation on the surface. This triangulation needs to be sufficiently fine to accurately resolve all details in the model, but the level of refinement also needs to be in parity with the resulting Cartesian grid, to avoid degraded accuracy in refined regions. Since the Cartesian grid is adaptively refined depending on the evolution of the solution, it is sometimes hard to predict how fine it needs to be and where the grid needs to be fine. The next step in the grid generation process is to validate the triangulation to make sure it represents a consistent water-tight surface without any self-intersecting triangles. The validated triangulation is an essential part of the input to the code Cubes (from the Cart3D package) which constructs an adaptively refined embedded boundary grid for a given grid size. This code saves its result in a file that is read by a preprocessor, which sets up the internal data structure used for the solver.

In the next 18 months, we envision a more seamless grid generation process, where the solver communicates directly with the grid generator through an API. After an initial (interactive) validation of the surface description, RAP would save the surfaces together with topological connectivity information in a data base. The embedded boundary grid would then be constructed through calls to the API. To ensure accurate intersection computations as well as fast responses, we plan to use an adaptively refined triangulation to represent the surface. In this way, it will be possible to gradually increase the resolution of the surface triangulation as the embedded boundary grid is refined. To enable this approach, it will be necessary to develop a new embedded boundary grid generator, replacing Cubes. The new grid generator will be tightly coupled to the geometry representation in RAP,

allowing the surface triangulation to be adaptively refined where needed. However, the surface triangulation would only be used internally in the grid generator, and it will no longer be necessary to explicitly save a triangulation, validate it, run Cubes, and finally convert the output from Cubes to the internal data structure used by the solver. Instead the solver will be able to directly query the grid generator through the API. Implementing the new grid generator will be non-trivial, but we anticipate that the resulting software will be worth the effort.

## 4 Advanced Algorithm Development

### 4.1 Advanced Methods for Complex Geometries

#### 4.1.1 Goals and Approach

In this part of the project, we are attempting to anticipate the development of more sophisticated approaches for solving PDEs in complex geometry by combining embedded boundary approaches for geometry representation with overset meshes and multi-element meshes with large regions of structure. To this end, we are performing research in improved algorithms and software for grid generation, and in discretization methods that take advantage of these new mesh types.

#### 4.1.2 Accomplishments

##### Representing Surface Geometry.

We developed algorithms to enhance the interactivity and automation of the embedded boundary and overset grid generation process for geometries defined by CAD descriptions [32, 20, 21]. In particular, we focused on the development of tools for the interactive generation of single component meshes from CAD data. In the overset grid and multi-element grid generation process, a set of these single component meshes is combined to produce a mesh that represents the entire geometry. CAD descriptions in terms of IGES file formats represent a complex surface as a set of 3D curvilinear patches each of which has a mathematical representation as a trimmed NURBS (non-uniform rational B-spline). A typical geometry can include large regions of smooth surfaces, sharp corners, and intricate details. The representation can therefore easily have hundreds or thousands of trimmed NURBS. When generating an overset grid, one would like to construct as few surface grids as possible while faithfully representing the intricate details of the geometry. Thus, tools for automatically dealing with the trimmed NURBS representation are essential for optimal user interactivity and efficiency.

To address this need, we implemented tools for interactively inspecting the mesh topology and verifying boundary conditions on a constructed mesh. We also implemented a capability for automatically generating the background Cartesian meshes that are used to fill volumes away from boundary surfaces. Tools for interactively specifying starting curves for the hyperbolic (marching) mesh generator, including the ability to easily specify a curve that crosses multiple patches in a CAD description, were implemented [19]. Features that help build higher-quality meshes, including ability to fit gridlines to a specified curve, and the ability

to march around sharp corners in the geometry were implemented. An interactive capability for clustering grid lines for local refinement was implemented, replacing parametric (non-GUI) approach in earlier versions of the software. An enhanced ability to automatically smooth surface grids and relax grid surfaces in regions of rapid geometry variation was implemented based on the solution of elliptic equations on 3D surfaces.

### Mapped Grids in 3D.

The goal of this work is the development of high-resolution finite volume methods on a general logically rectangular hexahedral mesh. This work was motivated in part by the desire of SciDAC users to use nonrectangular grids, in particular for fusion simulations in toroidal geometry.

There are two basic philosophies behind solving PDEs on a general mesh. In the first case, one can rely on the existence of an analytic mapping function which takes a uniform Cartesian mesh and maps it to a quadrilateral (in 2d) or hexahedral (in 3d) mesh and use this analytic function to transform the desired PDE. A second approach is to solve the original PDE, but modify the underlying finite difference/finite volume discretization of the PDE to accurately approximate derivatives in physical space on the non-uniform mapped grid.

We have chosen to go with the second approach. This approach is appealing because in the context of the wave propagation algorithm, one can easily rotate Cartesian data to align it with edges or faces of the mapped grid cells, solve Riemann problems in this rotated framework, and transform the data back. In our implementation of the above idea we made the assumption that, for example in 2d, the edges of each mapped cell are straight. In 3d, we assume that the faces of each cell form ruled surfaces. All terms relevant to our discretization of the PDE can be obtained by knowing only the location in physical space of the four (or eight) corners of the mesh cells of the mapped grid.

The geometric quantities that one needs include area (or volume) of the mapped cell, and length (or area) of each edge or face of the mapped cell. In 2d, it is straightforward to compute these quantities for a general quadrilateral cell with straight sides. In 3d it is less obvious how to approximate these quantities. In our approach, we approximate the faces of each cell using a trilinear map, defined from the eight vertices of the mesh cell. We then compute an orthogonal set of vectors, one normal and two tangent, to each face. We compute this basis at the center (in the uniform Cartesian coordinates) of each face. The volume of the hexahedral mesh is computed using an exact formula, and the surface areas of each face are approximated using a formula that is exact for planar faces.

We are now starting to test the 3d curvilinear code on standard Euler test cases. At this point, our tests have consisted mainly of comparing one dimensional problems in 3d to a 1d reference solution, and the results are quite promising.

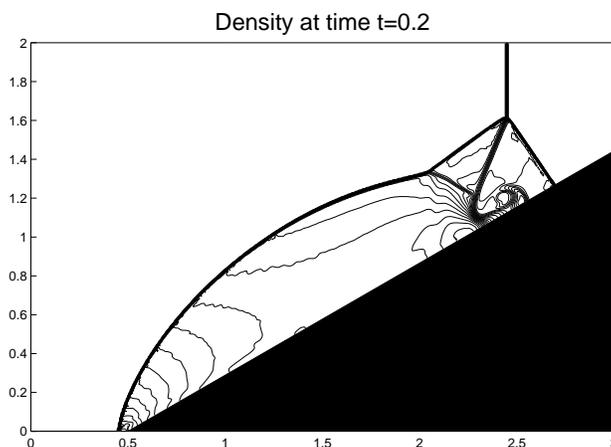
To extend this to AMR, we need to address the question of how accurate is our trilinear approximation to mesh cells when used in an AMR setting? By approximating curved cell edges with straight edges or non-planar faces by ruled-surfaces, we have no guarantee that a refined patch has the same area or volume as its underlying coarse patch. One consequence of this, in our implementation, is that interpolation between coarse and fine meshes is not formally first order accurate. Although the results of AMR computations look quite good, we plan to investigate this issue more carefully.

### Accurate discretization methods for embedded boundaries.

The focus of our work is the development of accurate numerical methods for Cartesian embedded boundary meshes. Several methods have been introduced over the last few years, in a variety of settings, for example for elliptic PDE [24], and steady state inviscid flow [1]. There are also some approaches for time dependent inviscid flow, using flux redistribution for stability [9], or using implicit methods [31]. However, in the explicit case these last methods do not retain second order accuracy simultaneously with maintaining stability for a CFL condition based on the spacing of the regular grid, away from the embedded geometry. In general, the question of accuracy on non-smooth meshes is not that well understood.

We completely analyzed a one-dimensional model problem on an irregular grid, developed a method that was second order on this grid, and proved its stability using the GKS theory. Stability is based on a cancellation property of the fluxes updating the small cell with  $volume \ll h$  (the 1-d volume). We also developed limiters for the approach, and showed examples with the Euler equations. The paper “H-box methods for the Approximation of Hyperbolic Conservation Laws on Irregular Grids” (Berger, Helzel, and Leveque) will appear in SINUM.

The two dimensional case is much more delicate. To maintain the cancellation property for stability, we use a rotated scheme at the embedded boundary. We first developed a second order rotated scheme on Cartesian grids for use at all cells. This scheme is an interesting generalization of an optimal scheme by Roe and Sidilkover. When applied at the boundary, the rotation angle is chosen to be normal to the embedded boundary in each cell. Again, limiters were developed. The figure below shows a computation of a ramp reflection using this new method. A manuscript is almost finished.



In the next year we intend to finish examining the two dimensional case, and if sufficient progress is made, extend it to three dimensions. The rotated scheme is expensive, entailing two Riemann problems per flux (in both the normal and tangential directions). We have ideas for maintaining stability using an operator split approach, which would reduce the work. Other plans are to investigate the time dependent case where the geometry is moving as well, and to examine the problems with limiters (both in this new scheme are more generally).

## 4.2 Higher-Order Methods

### 4.2.1 Goal and Approach

The goal of this work is the formulation, construction, and evaluation of higher-order spatial and temporal methods in computational fluid dynamics. The work is algorithmic in nature focusing on general issues of accuracy and efficiency in a wide class of problems governed by partial differential equations. Nevertheless the work is motivated by the applications of the APDEC proposal, specifically combustion research, and we have actively collaborated with application scientists.

Of particular interest to our effort is the open issue of how best to construct numerical methods for problems with multiple spatial *and* temporal scales. Much of the APDEC effort concerns the development of the numerical infrastructure necessary to apply advanced techniques in spatially adaptive numerical meshes to the application areas with multiple spatial scales. As this capability matures, it is natural to address issues of temporal accuracy as well.

### 4.2.2 Accomplishments

The bulk of the research effort thus far has been the development of new temporal integration techniques for advection-diffusion-reaction (ADR) systems. We have developed multi-implicit spectral deferred correction methods (MISDC) [8] for ADR systems. These methods allow the decoupling of the solution processes associated with the implicit treatment of diffusion and reaction terms, hence are similar to operator splitting methods. In addition, MISDC methods allow different time steps to be used with different processes. However, unlike operator splitting methods, MISDC methods are easily extended to higher-order accuracy.

In more recent work, a robust treatment of advective terms in conservation form, similar in style to the piecewise parabolic methods, has been incorporated into the MISDC method and applied to the reacting compressible Euler equations [25]. The resulting methods are up to fourth-order accurate in time and space in smooth regions of the flow, and robustly and accurately represent shocks.

In other work, Minion has constructed fourth-order adaptive methods for the 2-D Boussinesq equations which use deferred corrections and fast multipole accelerated solvers developed by Greengard, Huang, and Ethridge to invert the elliptic equations associated with diffusion and incompressibility [29, 28]. These methods combine higher-order, semi-implicit temporal integration based on semi-implicit spectral deferred corrections [30], with higher-order spatial methods based on the new versions of the fast multipole method with diagonal translation operators [18, 17]. In addition, most of the derivative approximations in the fluid solver are also computing using analytical formulas rather than finite differences, and hence are capable of extremely accurate solutions.

### 4.2.3 Future Plans

The plan for the next year is to move toward applying the new techniques which we have developed for fluid systems with multiple time-scales to real-world problems. Despite the

promise shown by the various spectral deferred correction algorithms, the work to date has been mostly algorithmic, and much needs to be done to impact the applications community.

The multi-implicit SDC methods have thus far only been applied to the reacting compressible Euler equations of gas dynamics in a one-dimensional setting. In order to apply the methodology to engineering applications, several issues must be considered. The extension of the methods to three dimensional problems is for the most part straight-forward; however, the treatment of non-trivial physical domains, time-dependent boundary conditions, and moving grids is more challenging. Accurate treatment of nontrivial boundary conditions is a difficult problem in most of the application areas of the APDEC project, and we hope to utilize the work already done by other APDEC investigators.

The second area of concentration in the coming year will be the development of higher-order temporal integration algorithms for the equations of low-Mach number reacting flow. These are the equations of interest in the applications pursued by Bell and Day. Unlike the reacting Euler equations which admit a straightforward operator split form, the low-Mach number asymptotic expansions used to model reacting flow contain elliptic constraints which couple all terms in the equations. In the higher-order projection methods for incompressible flow developed by Minion [29, 28] a non-standard auxiliary variable formulation is employed in order to achieve higher-order temporal accuracy in both the velocity and the pressure, but no similar extensions has been made in the low-Mach number setting.

## 5 Budget and Staffing

The current (FY 2003) budget for the APDEC project is \$2.7 M, with an additional \$80 K being provided by the SciDAC SAPP program. Of this, \$500K (\$100K each) goes to the five University PI's, with the remainder going to the three DOE laboratory participants (LBNL, LLNL, PPPL). We give a detailed breakdown of activities by site, and the names of all professional staff working on the project. Within the laboratory sites, we also give the division of the funding between applications, software, and algorithm development.

- **LBNL** : \$1.55 M = 6.76 FTE. Of this, 1 FTE goes to the Center for Computational Sciences and Engineering to fund the combustion application effort (John Bell, Marc Day). The remaining 5.75 FTE goes to the Applied Numerical Algorithms group. 1 FTE (David Serafini) is dedicated full-time to the accelerator modeling application, and 4.66 FTE are working on software development effort (Daniel Graves, Terry Ligocki, Peter McCorquodale, Brian van Straalen full-time, plus 1/3 FTE each of Ted Sternberg and Noel Keen). Phillip Colella, as lead PI, is funded at .1 FTE.
- **LLNL**: \$510K = 2 FTE. Of this, one FTE is dedicated to the development of embedded boundary grid generation tools, and one FTE to advanced gridding and discretization methods. Staff: Anders Petersson, William Henshaw, and Kyle Chand.
- **PPPL**: \$140K + \$80K SAPP funding = 1 FTE. This funds Ravi Samtaney to work on the magnetic fusion application.
- **New York Univ.**: \$100K. Prof. Berger and a postdoctoral researcher (Christiane Helzel during 2002; currently Lilia Krivodonova) work on embedded boundary grid generation issues and uniformly second-order accurate embedded boundary discretization methods.

- **Univ. of Washington:** \$100K. Prof. Leveque and a postdoctoral researcher (Donna Calhoun) work on high-resolution upwind methods on mapped AMR meshes in 3D, uniformly second-order accurate embedded boundary discretization methods, and interoperability between CLAWPACK and Chombo.
- **Univ. of North Carolina:** \$100K. Prof. Minion and a postdoctoral researcher (Anita Layton) work on higher-order spatial and temporal methods for fluid dynamics.
- **Univ. of California, Davis:** \$100K. Prof. Puckett and his students work on volume-of-fluid discretizations for fixed and free boundary problems.
- **Univ. of Wisconsin:** \$100K. Prof. Rutland and his students work on numerical methods for modeling sprays on AMR meshes.

In addition to the personnel funded directly by the project, we also have the following collaborators from the SciDAC applications projects and other ISICs.

- Steven Jardin, PPPL, and co-PI for the SciDAC CEMM project. Jardin has been working with APDEC on the development of AMR for MHD. evaluate results.
- Robert Ryne, LBNL, and co-PI of the SciDAC 21st Century Accelerator Science and Technology project. Ryne is one the developers of the IMPACT and MLI beam dynamics codes, and has been working with us on the coupling of AMR-PIC to these beam dynamics codes.
- Eric Esarey, LBNL. Esarey is a participant in the SciDAC 21st Century Accelerator Science and Technology project working in the area of plasma-wakefield accelerators. He has been working with us to develop the geometry tools required for the gas jet simulation.
- Jaideep Ray, SNL-CA. Ray is a participant in the SciDAC Computational Facility for Reacting Flow Science, and in the CCTTSS ISIC. He has been working with us on interoperability issues, including the AMR data alias and its application to the HDF5 interface, and is planning to collaborate on the development of the Chombo AMR elliptic solvers as CCA components.
- Robert Falgout, LLNL. Falgout is a participant in the TOPS ISIC, and has been working with us on the coupling of *hypra* to Chombo.

## References

- [1] M. Aftosmis, M. Berger, and G. Adomavicius. A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries. Technical Report AIAA-2000-0808, American Institute of Aeronautics and Astronautics, January 2000. presented at the AIAA 38th Aerospace Sciences Meeting, Reno, NV.
- [2] M. Aftosmis, J. Melton, and M. Berger. Robust and efficient Cartesian mesh generation for component-based geometry. *AIAA Journal*, 36(6):952–960, June 1998.
- [3] G.T. Balls and P. Colella. A finite difference domain decomposition method for solving Poisson’s equation using local corrections. *J. Comput. Phys.*, 180(1):25–53, 2002.

- [4] J. B. Bell, M. S. Day, and J. F. Grcar. Numerical simulation of turbulent methane combustion. *Proc. Combust. Inst.*, 29, 2002.
- [5] J. B. Bell, P. Colella, and H. M. Glaz. A second order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85(2):257–283, December 1989.
- [6] J. B. Bell, M. S. Day, J. F. Grcar, W. G. Bessler, C. Schultz, P. Glarborg, and A. D. Jensen. Detailed modeling and laser-induced fluorescence imaging of nitric oxide in an nh3-seeded non-premixed methane/air flame. *Proc. Comb. Inst.*, 29, 2002.
- [7] J. B. Bell, M. S. Day, J. F. Grcar, M. J. Lijewski, M. Johnson, R. K. Cheng, and I. G. Shepherd. Numerical simulation of a premixed turbulent V-flame. In *Proceedings of the 19th ICDEERS Conference*, Hakone, Japan. July 27–August 1, 2003.
- [8] A. Bourlioux, A. T. Layton, and M. Minion. Higher-order multi-implicit spectral deferred correction methods for problems of reacting flow. *J. Comput. Phys.*, 2002. submitted.
- [9] I.-L. Chern and P. Colella. A conservative front tracking method for hyperbolic conservation laws. Technical Report UCRL-97200, Lawrence Livermore National Laboratory, July 1987.
- [10] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, March 1990.
- [11] P. Colella, D. Graves, T. Ligocki, D. Martin, D. Modiano, D. Serafini, and B. Van Straalen. Chombo software package for AMR applications: design document. <http://davis.lbl.gov/apdec/designdocuments/chombodesign.pdf>.
- [12] P. Colella, D. Graves, T. Ligocki, D. Modiano, and B. Van Straalen. EBAMRGodunov design. <http://davis.lbl.gov/apdec/designdocuments/ebamrgodunov.pdf>.
- [13] P. Colella, D. Graves, T. Ligocki, D. Modiano, and B. Van Straalen. EBAMRTools:EBChombo’s adaptive refinement library. <http://davis.lbl.gov/apdec/designdocuments/ebamrtools.pdf>.
- [14] P. Colella, D. Graves, T. Ligocki, D. Modiano, and B. Van Straalen. EBChombo software package for Cartesian grid embedded boundary applications. <http://davis.lbl.gov/apdec/designdocuments/ebchombo.pdf>.
- [15] P. Colella, D. Graves, T. Ligocki, and B. Van Straalen. AMR Godunov unsplit algorithm design and implementation. <http://davis.lbl.gov/apdec/designdocuments/amrgodunov.pdf>.
- [16] W. Y. Crutchfield and M. Welcome. Object-oriented implementation of adaptive mesh refinement algorithms. *Scientific Programming*, 2(4):145–156, 1993.

- [17] F. Ethridge and L. Greengard. A new fast-multipole accelerated poisson solver in two dimensions. *J. Comput. Phys.*, 2000. submitted.
- [18] L. Greengard and J. F. Huang. Diagonal forms of translation operators for helmholtz and yukawa potentials. in preparation.
- [19] W. D. Henshaw. An algorithm for projecting points onto a patched CAD model. *Engineering with computers*, 18:265–273, 2002.
- [20] W. D. Henshaw. Generating composite overlapping grids on CAD geometries. In *8'th international conference on numerical grid generation in computational field simulations*, 2002.
- [21] W. D. Henshaw. An object-oriented framework for overlapping grid applications. In *Proceedings of the AIAA conference on applied aerodynamics*, 2002.
- [22] P. N. Hilfinger and P. Colella. FIDIL: A language for scientific programming. In Robert Grossman, editor, *Symbolic Computing: Applications to Scientific Computing*, Frontiers in Applied Mathematics, chapter 5, pages 97–138. SIAM, 1989.
- [23] R. A. James. The solution to Poisson's equation for isolated source distributions. *J. Comput. Phys.*, 25:71, 1977.
- [24] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 147(2):60–85, December 1998.
- [25] A. T. Layton and M. Minion. Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics. *J. Comput. Phys.*, 2002. submitted.
- [26] Peter McCorquodale. AMRNodeElliptic software package: node-centered AMR for elliptic problems. <http://davis.lbl.gov/apdec/designdocuments/amrnodeelliptic.pdf>.
- [27] M. Minion. On the stability of Godunov-projection methods for incompressible flow. *J. Comput. Phys.*, 123(2):435–449, 1996.
- [28] M. L. Minion. Higher-order semi-implicit projection methods. In M. Hafez, editor, *Numerical Simulations of Incompressible Flows: Proceedings of a conference held at Half Moon Bay, CA, June 18-20, 2001*. World Scientific Press, January 2003.
- [29] M. L. Minion. Spectral deferred corrections and projection methods. In *Innovative Time Integrators for Partial Differential Equations*, 2003. To appear.
- [30] Michael L. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. 2002. Submitted.
- [31] S. Murman, M. Aftosmis, and M. Berger. Implicit approaches for moving boundaries in a 3-d cartesian method. Technical Report AIAA-2003-1119, American Institute of Aeronautics and Astronautics, January 2003. presented at the AIAA 41st Aerospace Sciences Meeting, Reno, NV.

- [32] N. Anders Petersson. Overlapping grid generation on CAD models. In *6'th overset composite grid and solution technology symposium*, 2002.
- [33] N. Anders Petersson. A software demonstration of 'rap': preparing CAD geometries for overlapping grid generation. In *8'th international conference on numerical grid generation in computational field simulations*, 2002.
- [34] N. Anders Petersson and Kyle K. Chand. Detecting translation errors in CAD surfaces and preparing geometries for mesh generation. In *Proceedings of 10'th annual meshing round table*, 2001.
- [35] K. G. Powell, P. L. Roe, T. J. Linde, T. I Gombosi, and D. L. DeZeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *J. Comput. Phys.*, 154(2):284–309, 1999.
- [36] J. S. Saltzman. An unsplit 3d upwind method for hyperbolic conservation laws. *J. Comput. Phys.*, 115(1):153–168, 1994.
- [37] E. H. Twizell, A. B. Gumel, and M. A. Arigu. Second-order,  $L_0$ -stable methods for the heat equation with time-dependent boundary conditions. *Advances in Computational Mathematics*, 6(3):333–352, 1996.