LA-UR-04-0012

Approved for public release;
distribution is unlimited.

Title: Massively Parallel Simulations with DOE's ASCI Supercomputers: An Overview of the Los Alamos Crestone Project

Author(s): Robert P. Weaver and Michael L. Gittings

Submitted to: Chicago Adaptive Mesh Refinement Workshop Proceedings

Massively Parallel Simulations with DOE's ASCI Supercomputers: An Overview of the Los Alamos Crestone Project

Robert P. Weaver[1] and Michael L. Gittings[1,2]
rpw@lanl.gov and *gitting@lanl.gov*

[1] Applied Physics (X) Division, Los Alamos National Laboratory, Los Alamos, NM USA 87545
[2] Science Applications International Corporation, La Jolla, California 92121

Summary. The Los Alamos Crestone Project is part of the Department of Energy's (DOE) Accelerated Strategic Computing Initiative, or ASCI Program. The main goal of this software development project is to investigate the use of continuous adaptive mesh refinement (CAMR) techniques for application to problems of interest to the Laboratory. There are many code development efforts in the Crestone Project, both unclassified and classified codes. In this overview I will discuss the unclassified SAGE and the RAGE codes. The SAGE (SAIC adaptive grid Eulerian) code is a one-, two-, and three-dimensional multimaterial Eulerian massively parallel hydrodynamics code for use in solving a variety of high-deformation flow problems. The RAGE CAMR code is built from the SAGE code by adding various radiation packages, improved setup utilities and graphics packages and is used for problems in which radiation transport of energy is important. The goal of these massively-parallel versions of the codes is to run extremely large problems in a reasonable amount of calendar time. Our target is scalable performance to ~10,000 processors on a 1 billion CAMR computational cell problem that requires hundreds of variables per cell, multiple physics packages (e.g. radiation and hydrodynamics), and implicit matrix solves for each cycle. A general description of the RAGE code has been published in [1], [2], [3] and [4].

Currently, the largest simulations we do are three-dimensional, using around 500 million computation cells and running for literally months of calendar time using ~2000 processors. Current ASCI platforms range from several 3-teraOPS supercomputers to one 12-teraOPS machine at Lawrence Livermore National Laboratory, the White machine, and one 20-teraOPS machine installed at Los Alamos, the Q machine. Each machine is a system comprised of many component parts that must perform in unity for the successful run of these simulations. Key features of any massively parallel system include the processors, the disks, the interconnection between processors, the operating system, libraries for message passing and parallel I/O and other fundamental units of the system.

We will give an overview of the current status of the Crestone Project codes SAGE and RAGE. These codes are intended for general applications without tuning of algorithms or parameters. We have run a wide variety of physical applications

from millimeter-scale laboratory laser experiments to the multikilometer-scale asteroid impacts into the Pacific Ocean to parsec-scale galaxy formation. Examples of these simulations will be shown. The goal of our effort is to avoid ad hoc models and attempt to rely on first-principles physics. In addition to the large effort on developing parallel code physics packages, a substantial effort in the project is devoted to improving the computer science and software quality engineering (SQE) of the Project codes as well as a sizable effort on the verification and validation (V&V) of the resulting codes. Examples of these efforts for our project will be discussed.

1 Overview of the Department of Energy (DOE) ASCI Program

The ASCI program at the DOE was started in 1996 as part of the Department's Stockpile Stewardship Program (SSP). One of the main activities of the DOE National Laboratories is to maintain the safety and reliability of the United States nuclear stockpile in the absence of nuclear testing. For decades, the testing of our nuclear weapons allowed the weapons scientists to demonstrably certify to the national leadership that the weapons would perform as expected should the President require their use. Additionally, the weapons in the active stockpile now will be retained well beyond their expected lifetimes, and through surveillance of these weapons, we have found further complications caused by the aging of various materials in these systems. Without testing the effects of these changes, the scientists at the National Laboratories were given the difficult task of continuing to certify the safety and reliability of these systems. A variety of programs and initiatives were started to address this task. No individual program under SSP constitutes a "replacement" for experimental validation of a particular design (i.e. nuclear testing). Rather, the collection of programs (e.g. – Above Ground Experiments [AGEX] such as DAHRT and NIF and others; the Advanced Simulation and Computing Initiative [ASCI] for improved computational modeling; Enhanced Surveillance; etc.) constitutes the SSP program and allows the National Laboratories make the most informed decisions about the safety and reliability of the nuclear stockpile in the absence of testing.

One of the components of the SSP is the enhancement of the complex computing codes used to simulate the operation of a nuclear weapon. Complex codes have been used for decades in the design and analysis of nuclear weapons tests. Now, without testing to experimentally verify the proper behavior of these complex designs, it was decided to accelerate the development of the codes used to simulate weapons. Furthermore, as the systems in the nuclear stockpile age, the resulting material changes invariably result in significant three-dimensional issues that require verified and validated three-dimensional (3D) codes to evaluate and assess. The addition of the third physical dimension to the simulation codes results in a huge increase in the computer power and speed required in order to answer such 3D questions in reasonable

amounts of time. These requirements led to the ASCI Program at the DOE. The fundamental ideas involved in the ASCI program were to accelerate the development of high-end supercomputers and the associated (necessarily parallel) software in order to have approximately 100 TeraOperations per sec (TOps) capability by the year 2004. The ASCI program is a concerted collaboration among the National Laboratories, the computing hardware vendors, academic institutions and many contractors to provide the capability to simulate a model that has a billion computational cells and has an extremely complex geometry with multiple-nonlinear physics. And, by the way, this capability must be verified calculationally and validated against a wide variety of experimental results, both Above-Ground Experiments (AGEX) as well as previous nuclear tests that have been done. This effort is a daunting task to say the least. Some of what is presented here (the unclassified parts of the Crestone Project) documents the progress towards these goals.

1.1 An Evolution of ASCI Supercomputers

The ASCI supercomputers starting arriving at the DOE labs (Sandia, Livermore and Los Alamos) in ~1997, with one 3 TOps machine each at the three DOE labs. One of the fundamental characteristics of these machines was the massively-parallel structure. Each machine could be used as a single supercomputer with ~6000 processors available for use. At Sandia there was the ASCI Red machine, a MPP design with Intel chips. At Livermore and Los Alamos there were ASCI Blue machines (SMP architecture) with SGI building the Los Alamos Bluemountain supercomputer (6144 cpus) and IBM building the Livermore Blue Pacific machine. The first ASCI Level-1 mile-

Fig. 1. The Bluemountain 3TOps Los Alamos supercomputer.

stones were accomplished on these machines, each Laboratory fundamentally using its own resources to complete the milestones. I will have more on these results later. The next major step in the ASCI hardware was the delivery in 2001 of a 12 TOps ASCI White machine at Livermore, built by IBM on

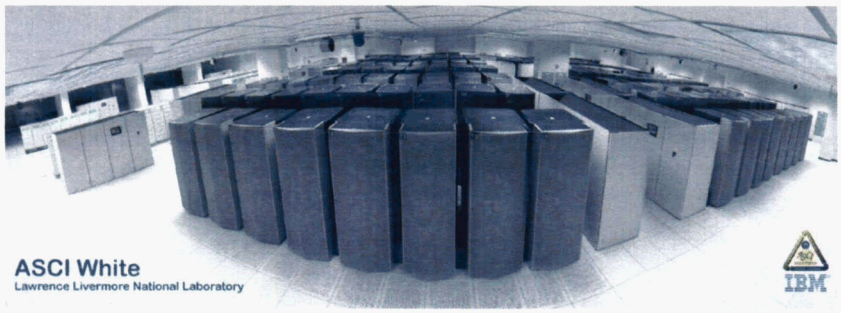

Fig. 2. The White 12TOps ASCI supercomputer at Livermore.

Fig. 3. The Q 20TOps ASCI supercomputer at Los Alamos.

the SP3 SMP architecture. Again many thousands of processors were available for use on single simulations, but this time the chip speed was greater, the interconnect was faster and the disk storage and I/O rates were greatly enhanced over the previous generation machines. The Global Parallel File System (GPFS) on the White machine holds 77 Terabytes (TB) of data! From the 3 TOps machines to the ASCI White 12 TOps machine, we actually saw a 3-4 fold increase in speed, or more to the point, a 4-fold decrease in the time-to-completion for a major 3D simulation. These machines were actually performing simulations that would have taken literally hundreds of years on any prior supercomputer, all due to the enhanced performance from parallel computations. The ASCI hardware acceleration continued in 2002 with the delivery of the ASCI Q machine at Los Alamos, a 20+ TOps capability supercomputer built by HP/Compact using alpha chips. Using about the same number of processors, this new machine could complete the same run $\sim$10 times or 3 times faster than either the Bluemountain or the White machine, respectively. So, not only were we running the same simulations faster, but we also could include more complex physics than before and run even more complex geometries than before. A complex 2D simulation that actually took

about one month to finish on the Bluemountain machine would run in about a week on the White machine and about two days on the Q machine. (So naturally we actually ran more more complex, detailed runs on the Q machine that finished in a few weeks time!) The next generation of ASCI supercomputers is still being worked, but a contract to IBM for a 100 TOps supercomputer at Livermore is scheduled for delivery in ~2006. I have refrained from discussing the details of the costs for these supercomputers. If interested one can contact the individual Laboratories for information.

1.2 An Evolution of ASCI Software – "Think Parallel"

With the advent of massively-parallel hardware available for general production use at the National Laboratories, there was a concerted effort to transition working codes and develop new codes that could take advantage of these resources. For the Los Alamos Crestone Project, this transition was a natural occurrence. The source code for the SAGE and RAGE codes that were delivered to Los Alamos from SAIC as part of the collaborative contract were initially written for the supercomputers of the early 1990's: a single vector processor architecture such as the CRAY machines. Since the capability to setup and run 3D simulations was available from the start of the contract (~1994), we soon realized that the supercomputers of the time (CRAY's) could not be effectively utilized for large 3D work. These machines were just not fast enough to complete 3D simulations (even rudimentary ones) in any reasonable amount of calendar time. From the 2D runs that were done in this time, we estimated that 3D runs would take more than 60 years to complete on the CRAY architecture. Thus, with the advent of the ASCI program in 1996, we were already starting a conversion process from the original CRAY vector Fortran 77 code to a new paradigm: parallel and modular code with effective SQE. There were several initial starts along this path, but the one chosen was to employ Fortran 95 with C program interfaces for I/O and the explicit use of message passing to effect communication among the processors. Domain decomposition was used to spread the computational cells of the problem among the processors. More details of this are given below and in some of the references ([3],[5],[6]). We needed to "Think Parallel" in our software design to begin running 3D simulations in a reasonable amount of time. For the "60 year" run to complete in just a month we would need an effective increase in computing power of 720 or almost 3 orders of magnitude! The "Think Parallel" phase is a reminder that simple conversions of algorithms developed for efficient use of CRAY vector single-processor architectures is much more that simply duplicating the computational mesh on each processor or even rearranging loops of the physics algorithms. In order to "Think Parallel" one must must completely change the framework and thought process (and hence the entire coding) for developing algorithms. An excellent, albeit brief, description of this thought process is given in a paper we wrote for the 1999 ISSW22 meeting in London [3].

2 An Overview of the Los Alamos Crestone Project

The Los Alamos Crestone Project developed from an extension of the highly successful collaboration between the scientists in the Thermonuclear Applications groups (X-2) of Los Alamos and the scientists at a major defense contractor, Science Applications International Corporation (SAIC). This high-performance collaboration team with members from both Los Alamos and SAIC continues today as one of the premier code-development successes at Los Alamos.

A little history helps put some of the important aspects of this collaboration in perspective. During the early 1990s one of us (Weaver) took an unusual Change-of-Station assignment (COS) for Los Alamos. Instead of the usual assignment of reporting to the DOE or DOD headquarters in Washington, DC, Weaver decided to perform a more technical COS and thus he moved to San Diego, California working on an enhanced collaboration for nuclear weapons effects that had been active for more than five years at that point. During this COS Weaver observed a code called SAGE, being developed by a remarkable individual Michael Gittings of SAIC. This code was originally developed to study the detailed effects of an underwater shock wave from a nuclear explosion at distances of many kilometers from the core of the detonation. Accomplishing the required resolution of centimeters to meters at distances of many kilometers from the source required a new approach to grid generation and grid development. The solution that Gittings was developing involved a continuous adaptive mesh refinement technique that was based upon a modern Eulerian hydrodynamics framework. In this method, one could refine any grid cell in the problem at any time in order to more accurately simulate the physics at that location. Conversely, if the physical gradients in a particular cell of the problem became small, then the method allowed the cells to become more coarsely resolved at that location. This dynamic controlling of the total number of cells in the problem results in minimizing the total memory and computer time required to complete that cycle of the problem, while maximizing the accuracy of the solution by having high-resolution in only those areas of the geometry that required them. We now call this technique Continuous Adaptive Mesh Refinement, or CAMR. Adaptive mesh refinement codes were not universally accepted in the early 1990s, but today the use of adaptive meshing seems to permeate a wide variety of computational physics from astrophysics to radiation transport to hydrodynamics, just to name a few. During this COS assignment in San Diego, Weaver decided to use Gittings' SAGE code to run some unclassified samples of problems of interest to the Laboratory. This simple exercise convinced us that creating a collaboration between SAIC and the Laboratory would result in a tremendous enhancement to the X-division existing code simulation capability. After the end of the COS assignment in 1993, X-2 initiated a contract with Gittings of SAIC that has continued through the present time. The results of this collaboration between a world-class code developer (Michael Gittings) and the

direction of an active code user has resulted in a truly amazing simulation capability for Los Alamos National Laboratory. This collaboration led directly to the creation of the ASCI sponsored Crestone Project at Los Alamos, in which the main goal of the project is to develop, verify and validate CAMR Eulerian hydrodynamics based capabilities for use in projects of interest to the laboratory.

Since the Crestone Project is funded by the DOE ASCI Program, there are many actual code development efforts managed by the Project. Some of the efforts, such as the SAGE, RAGE, NOBEL, SAGA and IAGO codes are unclassified codes that can be discussed in open literature such as this publication. Other efforts under the Project are classified and only some general characteristics of the truly amazing simulations that have been performed over the years can be described here. We believe it is an understatement to say that the Crestone project has far exceeded any expectations of the ASCI program. This fact is demonstrated by the projects continuous completion of many major Level-1 Milestones for the DOE ASCI program year-after-year (see below). More to the point, the examples of the use of the Crestone Project codes by many end-users has fundamentally changed the way the Laboratory scientists think about computing: massively-parallel simulations are answering questions though simulations that we never even dreamed of asking even as early as a few years ago. Here I will present several examples of these efforts. Other papers at this conference also highlight aspects of this statement; see, for example the work by Galen Gisler using SAGE and RAGE on complex 3D asteroid impact simulations. Although Verification and Validation (V&V) is a major activity of the project, the funding lines for V&V work are distinct from the basic code development work. Some examples of V&V for the Crestone project codes will be shown here, as well as many references for other examples([7],[4],[8],[10],[11],[12],[13],[2],[18],[23],[24]). The code continues to be used for a variety of application work, ranging from AGEX experiments, to volcanology, to fundamental physics of complex hydrodynamics ([9], [14], [15], [16], [17], [19], [20], [21], [22], [26], [27]).

The Crestone Project team currently is lead by three main individuals: Mike Gittings is the chief code architect for the project; Bill Archer is the Co-Project leader for code development and Bob Weaver is the Co-Project leader for physics design and initial demonstration applications. The FY2002 budget for the Crestone project was ~$7.2M and the FY2003 budget is ~$8.5M. This is a significant code development effort at Los Alamos and supports approximately 25 FTEs (full-time-equivalent staff). There are approximately two-three dozen users of the Project codes, both inside and outside the Laboratory. Although three-dozen users seems like a small number, for the kinds of specialized high-performance computing physics used in the Crestone Project codes, this number is actually larger than any other project we know.

The Crestone Project Unclassified Codes

The Fortran77 version of the SAGE code that Michael Gittings was developing in the late 1980's and the early 1990s had many of the capability elements of the current version of SAGE: 1D, 2D or 3D geometry, CAMR Eulerian based high-order Godonov hydrodynamics([28],[29]), and adaptive mesh capabilities optimized for hydrodynamic shock problems. Today's version of SAGE does all these things as well, but has been completely re-written with a massively-parallel implementation based on Fortran 95 and the message-passing interface (MPI).

The RAGE code is built some the SAGE code base by add a variety of physics, most notably, a two-temperature radiation diffusion package (see [1]). Two key individuals responsible for the development and implementation of the radiation diffusion solver in RAGE are Tom Betlach and R. Nelson Byrne, both of SAIC. They have not only written the original version of the implicit diffusion solver, but have also maintained this part of the code in production mode since its beginning. They both interact (almost daily) with end-users in order solve problems as they arise with this package. Additionally, Tom Betlach is aggressively pursuing modern options for implementation of multi-grid methods in this section of the code physics (see below). The main characteristics of the RAGE code are: multi-material Eulerian CAMR in 1D, 2D or 3D; a modern treatment of the Eulerian hydrodynamics based originally on the Piecewise-Parabolic Method (PPM) used in astrophysics since the early 1980's([28],[29]); unit aspect ratio cells (square in 2D and cube in 3D); radiation energy transport by a two-temperature (material and radiation) grey diffusion solution; preliminary interface treatment; and a material strength package. The SAGE and RAGE codes clearly have joint intellectual property rights between SAIC and Los Alamos. The CAMR adaption algorithms are the heart of the methodology and are implemented so that any two adjacent cells can differ by no more than a factor of two in mesh size. Each level of refinement is accomplished by halving the size of the mother cell in each direction, so the factor of two requirement between adjacent cells is equivalent to one level of adaption. The implicit radiation diffusion solve uses a conjugate gradient iteration with a point Jacobi (diagonal scaling) preconditioner. There is a variety of efforts underway currently to employ multi-grid techniques to the CAMR Eulerian mesh for the radiation diffusion solver. A couple of these efforts show great promise, where the payoff is measured in an actual reduction in the wall clock time to complete a full cycle of the radiation hydrodynamics. Merely reducing the iteration count is, in our opinion, not the only measure of success, while reducing the overall computation time is the desired result.

The parallelization strategy for these codes is based on the use of Fortran 95 and the message passing interface, or MPI, paradigm. Both allow for portability and scalability. In fact, both SAGE and RAGE are used on all available supercomputer platforms and are even used in the evaluation of new machines. The codes are so portable that they even run desktop and laptop

computers, including the Apple Macintosh with MacOS X unix environment. [In fact, most of the development work is done on the Mac and then code versions are immediately uploaded and used on 1000 processor supercomputers.] Load leveling among the many processors used in a simulation is based upon a simple reordering of the cell pointer list at the end of each computational cycle. At each cycle of the simulation a determination is made for each cell in the problem as to whether to subdivide a cell (refine the zoning) or merge adjacent cells (coarsen zoning). This decision is based upon the magnitude of the local physical gradients at that location. If the gradients become steep, then the code creates finer zoned daughter cells from coarser cell meshes; alternatively when the gradients become small enough then daughter cells are recombined into the original mother cells to produce a coarser mesh. The load leveling of a massively parallel run is maintained effectively by inserting the newly created daughter cells immediately after the mother cell in the global cell list, thereby maximizing the probability of on processor memory locations. The global cell list is reorganized at each time step (M total new cells at the end of a time step on N processors results in M/N cells per processor) and the result is excellent load-leveling. Empirically we have found that good performance is obtained with cells-per-processor in the range of 10,000 - 100,000.

Many finite difference codes use computational grids that are logically rectangular and structured. In this type of code the finite difference equations are often coded as follows:

$$something(i,j) = (a(i+1,j) - a(i-1,j))/dx(i).$$

Parallelization of this type of code structure is difficult, and typically involves the use of ghost cells (at the domain boundaries). This code structure also limits the problems to the number of dimensions that are hard-wired (two dimensions in this example). The philosophy used for the CRAY vector version of the RAGE code was different. This same expression in RAGE would be:

$$something(l) = (a(l\_right(l)) - a(l\_left(l))/dx(l),$$

where l_right(l) and l_left(l) are the global addresses of the cells to the right and the left of cell l. This method uses more memory and is somewhat slower than the original example, but it makes the coding for unstructured grids much easier. An added benefit is that it allows the same code to do 1D, 2D and 3D problems by looping over the dimensions. In order to parallelize this concept, our main concern was to have a code that scaled well. The resulting implementation uses message passing to obtain a local *copy* of the neighboring cell's data. The essence of this method is the creation of a local scratch array of values corresponding to the data needed from the cells to the right and left (up and down; near and far) of the current cell. The current version of RAGE uses the following structure to parallelize this example (where NUMDIM is the number of dimensions in the problem):

$Do\ DIR = 1, NUMDIM$
call $get\_next(DIR, HI\_SIDE, a, a\_hi)$
call $get\_next(DIR, LO\_SIDE, a, a\_lo)$

$something(l) = (a\_hi(l)) - a\_lo(l))/celldimension(l, DIR)$
endDo

The subroutine *get_next* obtains a copy of the requested cell-centered data ("a") from an array which can located anywhere (i.e. any processor). All communications between processors is hidden from the user and the resulting scaling is very good. This structure allows the code to be very portable, relying only on the machine having the basic MPI libraries. More details on these techniques are given in reference [3].

Since gather/scatter MPI routines are used to copy required data from where it resides (i.e. - which processors memory) to the local processor memory, cache reuse is actually quite good in the parallel implementation. A very nice report of models of the scalability of the SAGE code is given by Kerbyson, et.al. at Supercomputing 2001 and Supercomputing 2003 ([5],[6]). The primary goal of the parallel implementation is maintainability, scalability and portability to new platforms on problems of 1 billion CAMR cells. Maintainability refers to the suite of software quality practices in the Crestone Project in order to produce understandable, modular code that has a well-documented and reproducible heritage. Since ASCI hardware, and indeed the computing industry itself, seems to be changing hardware every 1 or 2 years, the fundamental goal for the structure of our project codes is portability with scalability to 1000s of processors. Ideal scaling in this sense would be to run a problem on 10000 processors (with 10000 times the workload [i.e. cells]) in exactly the same wall-clock time it takes to run on 1 processor with 1/10000th the work. Although we have had much greater success with scalability on MPP based machines, such as the ASCI Red machine or the Cray T3E, the scaling on SMP based machines is reasonably acceptable. Fig. 4 shows a graph of the scaling results on a wide variety of supercomputers.

2.1 Crestone Project Codes Represent an Improvement Over Previous Generation Techniques

Most of the computational work performed in X-division at Los Alamos, as well as in the design divisions at Livermore, has traditionally relied on pure Lagrangian hydrodynamic methods. There are two substantial problem areas from past 2D Lagrangian code use that the Crestone Project codes solve. Speaking from the perspective of an end-user, the first major difficulty for a Lagrangian hydrodynamics based code is that of setup, or generating the logical and physical mesh with the constraint of having complex geometry following contour interfaces. The second major problem historically with 2D Lagrangian based codes is that of mesh tangling as the simulation proceeds. If there is even a slight aspect of complex hydrodynamics involved in the simulation, such as vorticity or sheer flows, then a major portion of time has been spent in actual hand manipulation of mesh points (manual rezoning) to continue the simulation. The tools and codes in the Crestone Project solve both of these problems in a elegant fashion. We should mention that an alternate

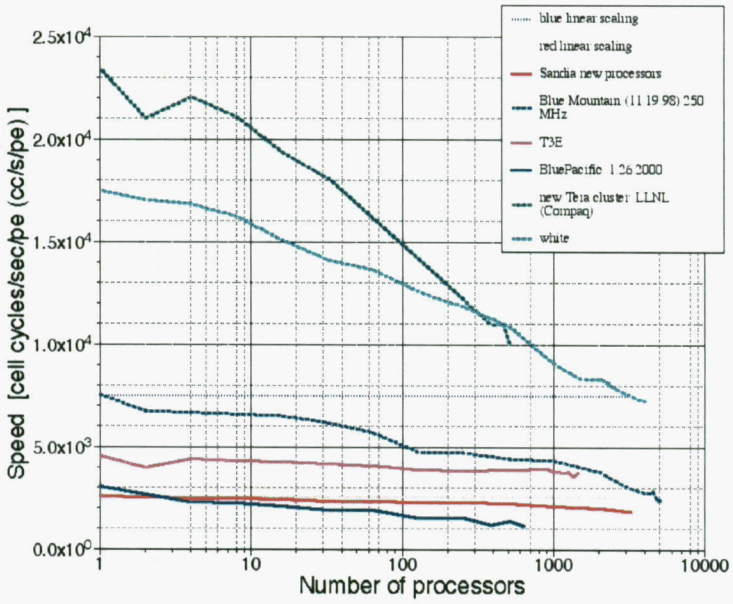

Fig. 4. The scaling performance of the SAGE code on various supercomputers.

path to a modern solution to the 2D Lagrangian problems is accomplished by the use of Arbitrary Eulerian-Lagrangian, or ALE hydrodynamic algorithms. This hydrodynamic framework is, in fact, used by the other major ASCI code-development efforts at both Los Alamos and Livermore. As with any complex multi-physics code projects, there are advantages and disadvantages of any particular formulation.There are strengths and weaknesses of both the CAMR Eulerian approach, as well as the ALE approach. However, for current ALE schemes, complex geometry grid generation as well as mesh-motion both remain difficult and research oriented aspects of those efforts.

2.2 Problem Setup for Complex 3D Geometries

Although there are mesh generation tools built into the SAGE code, by far the simplest approach to grid generation is by importing 3D solid model representations of the object being simulated. Mainly through the work of Rob Oakes and his co-workers in X-division of Los Alamos([25]), we have the ability to import nearly any 3D solid model geometry into the code and build a multimaterial mesh based on this geometry. The beauty of this whole process is that once the solid model exists, then the code itself parses the geometry and automatically creates the CAMR grid conforming to the input geometry. The end-user needs only to specify the physical size of resolution required for capturing the appropriate physics in each material. The code does the rest! Examples of this setup are shown in Figs. 5, 6 and 7.

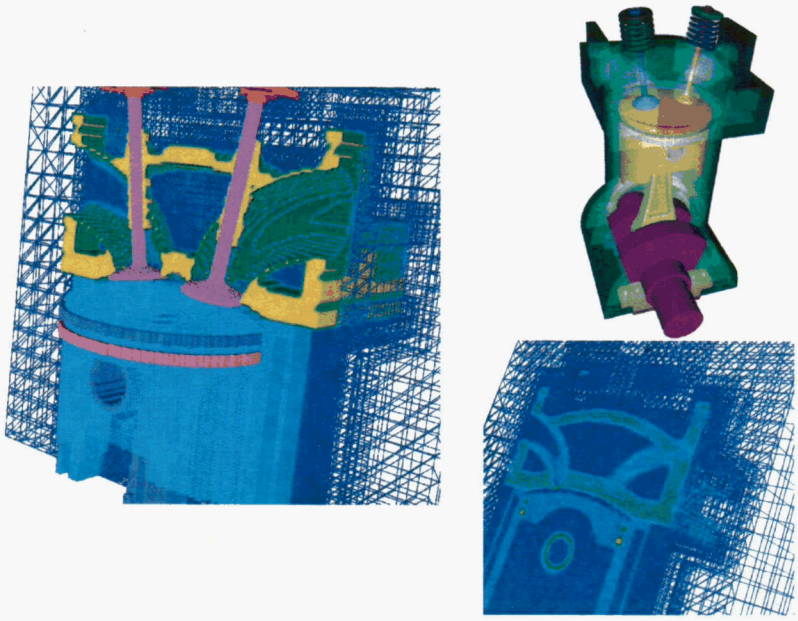

Fig. 5. An example of the use of CAD/CAM 3D solid geometry modeling to generate a continuous adaptive mesh refinement mesh for a portion of a piston engine.

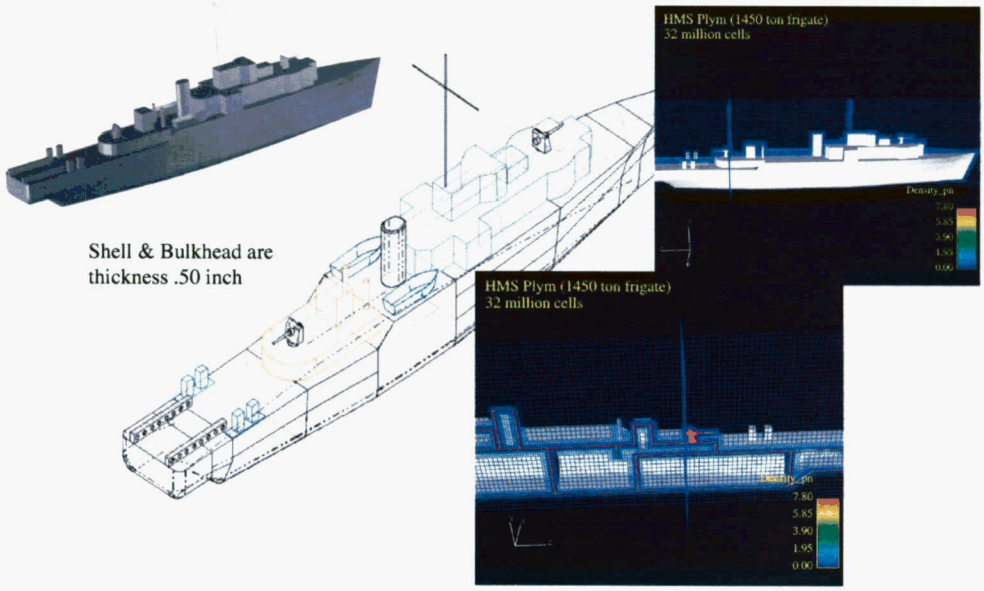

Fig. 6. An example of the use of CAD/CAM 3D solid geometry modeling to generate a continuous adaptive mesh refinement mesh for a portion of a British River-class Frigate.

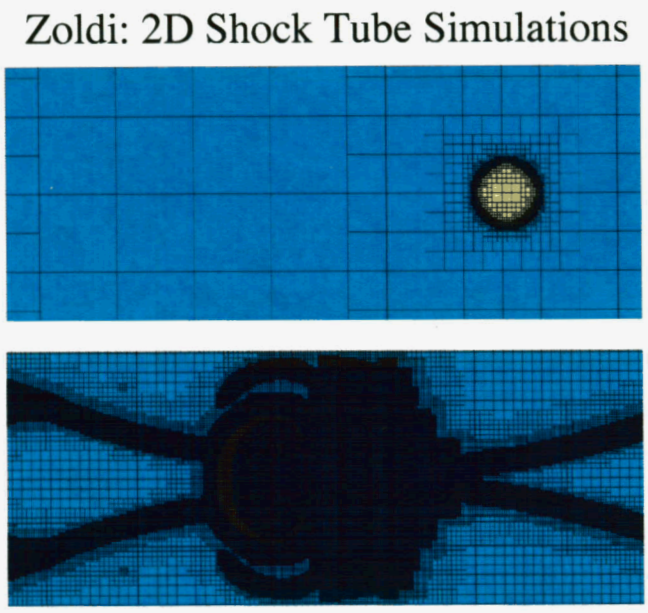

Fig. 7. An example of the continuous adaptive mesh refinement mesh for a simulation of a shock tube experiment involving a cylinder of SF_6 gas in air.

2.3 Continuous Adaptive Mesh Refinement (Cell-Based Refinement)

The second traditional problem of Lagrangian based hydrodynamics from the past has been the need for hand rezoning of complex simulations in order to prevent time-step crashes and/or badly formed zoned (e.g. bowties or boomerangs; or worse yet negative volume cells). This activity traditionally dominates an end-users time in completing a 2D complex simulation. The continuous adaption logic built into the Crestone Project codes allows the code to do the work as the problem progresses with time. Cells in the simulation are placed (created or re-combined) in the 1D, 2D or 3D geometry where they will provide the most accurate representation of the physics. This continuous adaption in both space and time is performed every cycle of the simulation. The overhead associated with this CAMR technique has been empirically determined to be ~20% of the total runtime. Since we gain several orders of magnitude in efficiency by the use of CAMR (over uniform meshing), this overhead time is completely acceptable. The true breakthrough for the end-user is the time saved both in complex problem setup as well as with *no* hand-rezoning required.

Currently, the adaption algorithms in the Crestone Project codes are based independently on two criteria: one level of resolution can be set to resolve material interfaces, while a different level of resolution can be set for the

physics within a material. A simple example as shown in Fig. 7 would be to have the grid resolution between the SF_6 cylinder and the air be followed dynamically with say 0.025 cm cells, while the shocks in the air (high-pressure gradients) can be followed dynamically with 0.050 cm cells and those shocks in the SF_6 material can be followed with 0.0125 cm cells. All of these criteria are used in a dynamic sense, so that cells are created (daughter cells) and cells are coarsened into original mother cells at each time step at each cell in the problem.

A dramatic example of the 3D CAMR method is shown in Fig. 8. This Figure shows a 3D simulation of a shock-generated instability (Richtmyer-Meshkov Instability [RMI]) from the passage of a mach 1.2 shock over a perturbed surface of SF_6 in air. The initial perturbation is in the form of a cosine-cosine distribution and this figure represents a time at which the interface between the air and SF_6 has been shocked from right-to-left and then reshocked from left-to-right in the Figure. Notice the extremely high resolution in the simulation that defines the complex interface between the two materials.

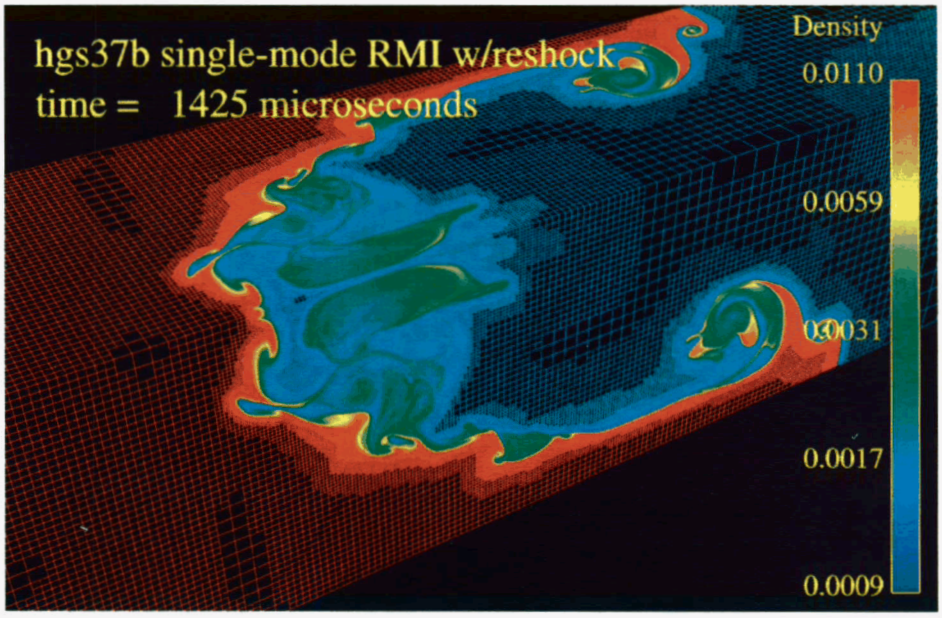

Fig. 8. An example of the 3D CAMR mesh for a simulation of a shock tube experiment involving a perturbed surface of SF_6 gas in air.

3 Examples of Massively-Parallel Simulations by the Crestone Project

In this section we will show several examples of simulations done with the Crestone project codes. The results will be shown chronologically in order to emphasis the success of the portability and scalability of the software framework. We show an evolution from one of the first generation ASCI machines (the 3 TOps Bluemountain supercomputer) to the 12 TOps White machine at Livermore and finally to the 20 TOps Q machine at Los Alamos. Remember our goal is to have a ~1000-fold decrease in wall clock time for 3D runs (from single processor run-times).

Although some initial 3D runs were performed on the CRAY computers in 1995 and 1996, the real onset of production 3D simulations with the Crestone Project codes started with the delivery of the Bluemountain supercomputer at Los Alamos in 1977. Some of the first runs that we did were related to hydrodynamic mixing of two materials. In 1998, we did a demonstration run on the Bluemountain machine of the 3D Rayleigh-Taylor instability (RTI). A graphic from this run is shown in Fig 9. With only four levels of refinement and 310 processors we were able to finish a simulation in 360 hours of cpu-time or about 15 days of continuous computing. The actual wall-clock time required to complete this run was about one month of calendar time. A comparison we will make as a standard measure of the parallel performance of these runs is the cpu-equivalent time required to run the simulation on a single processor of the same machine. So for this 310 processor run of 360 hours the single processor time would have been ~13 years! The cell-count of this simulation varied due to the CAMR with time because the surface area of the interface between the two mixing fluids grew larger with time. The problem started with about 7 million cells and ended with about 45 million cells, effectively running on only 310 processors.

The next step in the evolution of Crestone Project 3D parallel simulations was the RMI mixing simulation shown earlier in Fig. 8. This run was modeling the non-linear evolution of RMI from a mach 1.2 shock crossing a perturbed interface between air and SF_6 gas. Another view of this simulation is shown in Fig. 10. These runs were performed in 1999 after the full Bluemountain machine had been delivered to Los Alamos. This full machine of 6144 processors allowed us to perform a parameter study for the same initial conditions on mesh resolution (minimum cell size or maximum number of levels of refinement). In this study we ran the same 3D problem with six, seven and eight levels of refinement successively. The level six run ran to a problem time of 1.8 ms in 166 cpu-hours or 4.8 days of continuous run time. A summary of the three runs is shown in Table 1. The Level 6, Level 7, Level 8 runs were equivalent to a 2 year, 52 year, and 239 year single processor runs (respectively), while they were actually completed in 4.8 days, 34.5 days and 91 days, respectively. So the most refined run, the Level 8 study, was run in 3 months but would have taken over two centuries to complete on a single

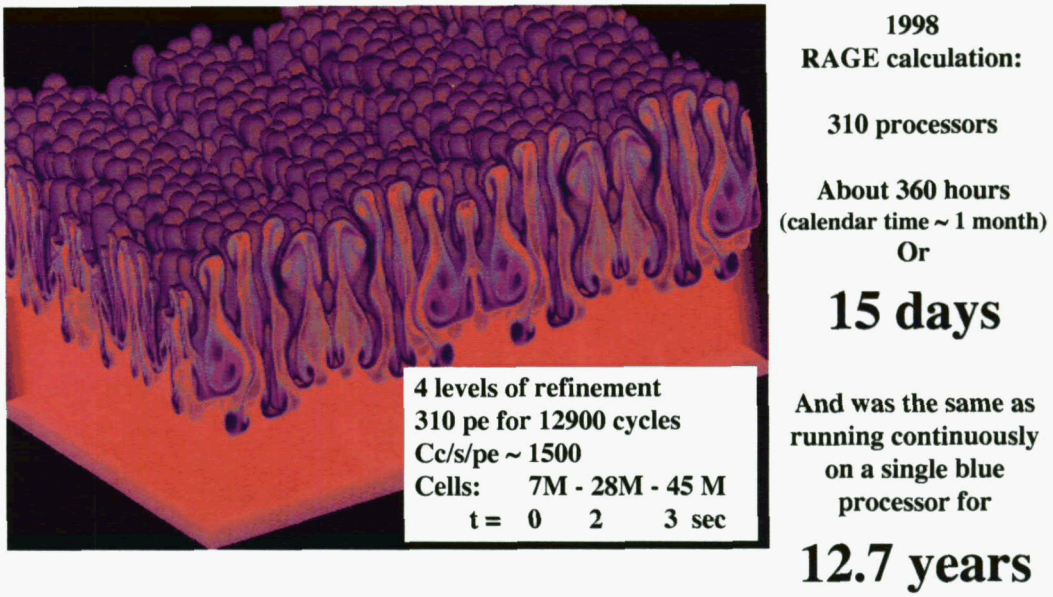

Fig. 9. The first large scale 3D simulation performed with SAGE on the Bluemountain machine: a multimode simulation of Rayleigh-Taylor mixing of two fluids of differing densities in a gravity field.

Table 1. Parameter Study on Mesh Resolution for a 3D RMI Problem

Level 6	Level 7	Level 8
CPU-hours	Processors	Max Cells
116	126	15M
828	1260	98M
2187	1890	181M

processor machine (assuming, of course that there would be enough memory!) These kinds of numbers clearly demonstrate the success of the parallel implementation of the SAGE code and the power of parallel computing. None of these runs would ever have even been started prior to the parallel hardware and parallel software. Within just two years of the start of the ASCI program, the National Laboratory design community was beginning to believe that massively-parallel computing would fundamentally change the scope of simulations that were done.

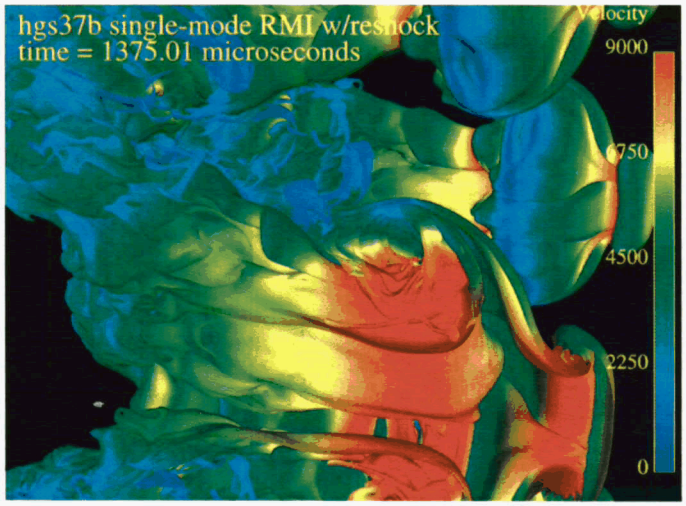

Fig. 10. The second large scale 3D simulation performed with SAGE on the Bluemountain machine: a single mode simulation of RMI mixing of SF_6 gas in air. This visualization is from the Ensight commercial graphics software from the run with seven levels of refinement.

3.1 ASCI Milepost Simulations 1999-2001

In this section we will take a short digression from the unclassified work that has been described in previous sections. As we stated earlier, there are both unclassified and classified code development efforts within the Crestone Project. Although we cannot describe any of the physics or algorithms used for the classified codes, we can summarize the magnitude of the simulations that have been performed. All of the information contained here is unclassified and has been reported in a wide variety of unclassified venues, such as Laboratory Press Releases, newspaper articles and DOE sponsored reports. A significant aspect of the initial phases of the ASCI program was the accomplishment of a series of major 3D simulations, known as ASCI milepost runs, that would demonstrate a progression of capability as the parallel software matured and as the size of the ASCI parallel platforms increased. From 1999 - 2001 there were very clearly defined, major 3D runs that each Laboratory was expected to complete on time. A Blue Ribbon panel of experts was formed (the Burn Code Review committee) to review the work done and determine whether the runs that had been completed actually satisfied the requirements. In each case, these initial milepost runs were single large scale 3D simulations which were to be compared to data from actual nuclear tests. It is an understatement to say that these milepost runs were extremely taxing on the resources available as well as the personnel who developed the software and those who performed the runs. We will get back to the first of these milepost runs (1999) in a moment.

The 2000 Milepost Simulation: A 3D Secondary Prototype Performance Simulation

The first milepost simulation that the Crestone team was scheduled to perform was the 2000 3D Prototype Secondary simulation. Although there are unclassified details of the requirements for this task, we will not describe those here. Instead, we would like to continue the documentation of the success of massively-parallel simulations performed by our team by detailing the statistics of these milepost simulations. This single run was performed on the Bluemountain 3 TOps machine using about 1/3 of the processors. The typical run would use 2016 processors: 126 of the 128 processors per SMP box (or node) and 16 SMP boxes. This arrangement would leave two processors idle to process system level requirements. The 3D secondary simulation we performed used ~100 million cells and ran over the course of a three month period. It consumed 2 million CPU hours, or the equivalent of a 230 year single processor run! Each processor of the Bluemountain machine has 0.25 gigabytes (Gb) of memory and this simulation did not stress the memory usage. The main simulation done created 15 terabytes of data. The visualization of the run was done with the Ensight commercial software package. During this timeframe (~2000) the Ensight code was serial, and it literally required the equivalent of 2 man-years to create a single 100 frame movie of the run. This process was painful and led to the high priority task of parallelizing the graphics software. With this simulation, as well as the one performed at Level 8 for the RMI 3D simulation, we had achieved our goal of a thousand fold increase in computing power. This 2000 milepost simulation was completed in April 2000, just before the horrendous firestorm in Los Alamos. Since the run was not due to be complete until the end of the calendar year, we successfully completed the task a full nine months ahead of schedule! In the remaining time before the year-end, we actually completed another full sequence of a 3D simulation for another test. These results were all documented and provided to the Burn-Code Review committee in January 2001. The milepost runs we had performed were judged to completely satisfy the requirements.

Towards the end of the year 2000, the Crestone Project team was called upon by the Los Alamos management to help with the completion of the original ASCI 3D performance milestone, the 1999 3D primary prototype simulation. This original milepost run had been successfully completed by Livermore, but still was not complete at Los Alamos in late 2000. After finishing our scheduled milepost run well ahead of schedule, we were in a position to help complete the 1999 milestone run. This was done towards the end of the year 2000 as a joint effort from the Crestone Project team and the Shavano Project team of Los Alamos.

The 2001 Milepost Simulation: A 3D Full-System Prototype Performance Simulation

Following directly on the heels of the completion of the 1999 and the 2000 ASCI Level-1 milepost simulations, the Crestone Project team was again scheduled to perform the next ASCI milepost: a prototype full-system simulation, which was due by the end of the calendar year 2001. Early in 2001, the White 12 TOps machine became available for use by our team. This White machine was physically located at Livermore, California and was connected to Sandia National Laboratory and Los Alamos National Laboratory by a high-bandwidth, secure Wide-Area Network (WAN). The project codes were moved to the White machine, compiled and tested. Since we had built in portability by our choice of software, this process of obtaining a production capability on White took only a few weeks. Much of the credit for handling the details of this transition go to Bob Boland, who spent much time living in Livermore, working with the system staff on the White machine. This milepost simulation, a 3D full-system prototype run, was (and still is) the most challenging task we have undertaken in our careers. This type of simulation had never been done before, either in 2D or in 3D, so we were not confident of a scientifically viable solution to this task almost until the run was complete! We did many of the preliminary simulations in 2D in order to demonstrate feasibility of the full-system concept with much shorter runs. This work of 2001 stands out as one of the Crestone Project team's most significant tasks. The statistics from this run are truly stunning. The main 3D simulation was done on the state-of-the-art (for it's time) ASCI machine, the White machine at Livermore, using Los Alamos's entire allocation of processors – 2048. The White machine is comprised of SMP nodes with 16 processors per node. The Los Alamos allocation on this machine was 128 nodes. Thus, this simulation alone used ~3 TOps of computing power. This single run took nine months to complete. It was started in January of 2001 and finished in October 2001. There was actually a couple of months during the summer 2001 that were devoted to additional code-development, so the the actual calendar time for this run was ~7 months. The actual day-to-day runs were performed in a sequence of runs that produced check-point restart files to start the next run of the sequence. Each of these individual runs was submitted for 96 hours, and many of the runs actually completed the full 96 hour allotment. Many, however, terminated prematurely for literally hundreds of differing reasons. With each terminated run, we would back up to the previously written restart file and continue. This one run consumed over 6 million cpu hours and is equivalent to a single-processor run that would take over 700 years to complete! With the associated 2D and other 3D simulations that went into this effort, we are calling this run our "millennium" simulation. The maximum cell count used was ~500 million cells. Restart files ranged in size (depending on the physics) from ~100 Gb to 240 Gb each. The wall-clock time to run one cycle ranged from 4 minutes to ~15 minutes. The parallel I/O on the White machine was

impressive: average numbers for this run are ~1 Gb/sec writes and ~2 Gb/sec reads, so reading a 240 Gb restart file took only ~2 minutes. The total data written from this run amounted to over 200 terabytes (Tb), with about 20 Tb devoted to Ensight graphics files. The size of the Library of Congress is about the same.

In addition to these amazing statistics, one should realize that this entire process was carried out *remotely*. The runs were actually done at Livermore, while we resided in Los Alamos, NM. Although it is fact of life these days to run remotely on computers around the world, the truly amazing accomplishment was the ability to manipulate and visualize the extremely large data sets involved. By this time (2001) we had a parallel version of the Ensight client-server commercial graphics software (called the server-of-servers [SoS]) that domain decomposed the geometric mesh into blocks (typically 50 - 64) of ~5 million cells each. With this software, we were able to leave the large data-sets on the White GPFS, use the SoS on several White nodes devoted to visualization, and display the graphics back at Los Alamos on a workstation running the Ensight Client software. This process worked well. We essentially had real-time graphics from our runs a thousand miles away, without the need to move terabytes of data around the country. In fact, the White GPFS was the only file system available that had the capacity to hold these large data sets. This simulation was presented to the Burn-code review panel in January 2002 and was judged to meet or exceed all of the requirements of the milepost. Our thanks go out to all who help make this effort successful, from the Project team members, to the consultants at Los Alamos, to the staff at the Livermore Computing center.

We think it is very clear from this progression of massively-parallel computing simulations, that the concept of parallel computing has been extremely successful. In 1996 at the start of the ASCI program, we could not find many in our community that would have believed this statement.

3.2 Current Massively-Parallel Simulations

Since the completion of the 2001 ASCI milepost simulation, the definition of these mileposts, and the overwhelming dedication required to perform them, has been scaled back. Current ASCI milestones are more directed at programmatic day-to-day work, as opposed to the performance of a "demonstration" simulation. We learned a lot from the large scale runs we performed, but they also take a significant toll on the personnel involved. Although large-scale 3D simulations continue today, the complexity and the computer science required has not changed much since 2001. Of note, however, is the delivery of the ASCI Q (HP/Compaq alpha processor) machine at Los Alamos in 2002. During the delivery of this machine, a 10 TOps section was devoted to unclassified science runs. One of our team members, Galen Gisler, applied for, and was granted time on this machine for a science run. Gisler's work is featured at this conference and we would direct interested readers to his work ([26],[27]).

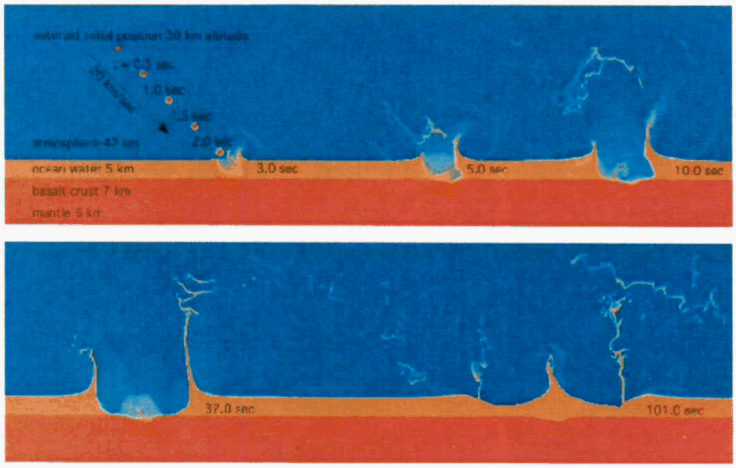

Fig. 11. A montage of 10 separate images from the 3-d run of the impact of a 1-km iron bolide at an angle of 45 degrees with an ocean of 5-km depth. These are density raster graphics in a two-dimensional slice in the vertical plane containing the asteroid trajectory. Note the initial uprange-downrange asymmetry and its disappearance in time. Maximum transient crater diameter of 25 km is achieved at about 35 seconds. The maximum crown height reaches 30 km, and the jet seen forming in the last frame eventually approaches 60 km.

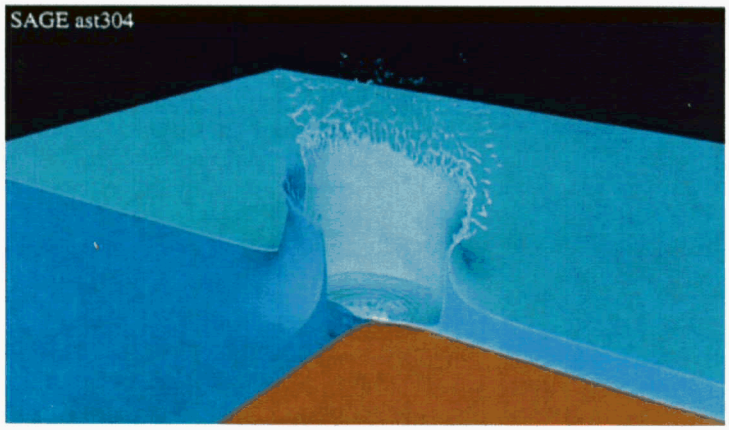

Fig. 12. Perspective plot of three isosurfaces of the density from the 3-d run of a 45-degree impact of a 1-km iron bolide into an ocean of 5-km depth at a time 30 seconds after the beginning of the calculation (27.5 seconds after impact). The isosurfaces are chosen so as to show the basalt underlayment, the bulk of the ocean water, and cells containing water spray (mixed air and water). The asymmetry of the crown splash is evident, as is its instability to fragmentation. Cratering in the basalt is seen, to a depth of 1 km. The diameter of the transient cavity is at this time 25 km.

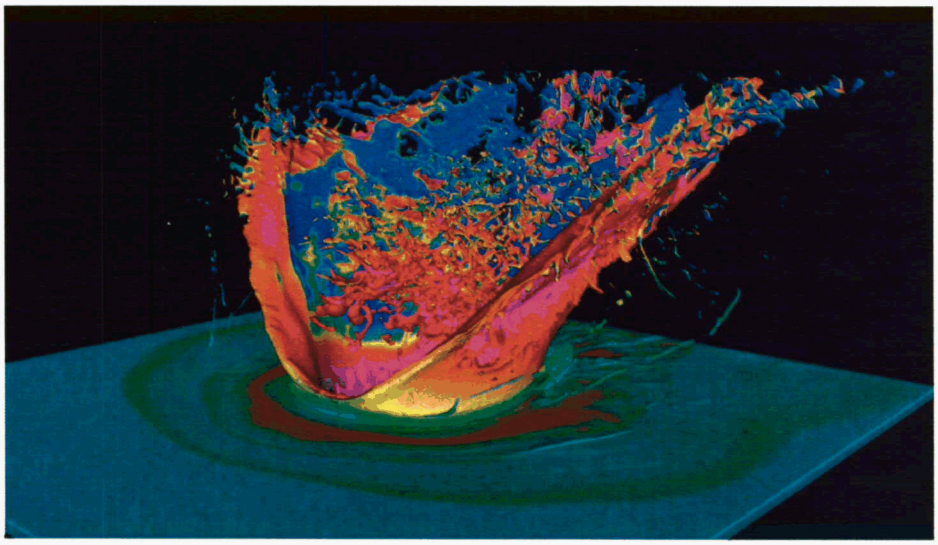

Fig. 13. A simulation of the impact crater at the Chicxulub site in the Yucatan Peninsula of Mexico. Forty-two seconds after impact, the rooster tail has left the simulation volume and gone far downrange. The image shows the extent of the calculational mesh: 256 km on a side. The dissipation of the asteroids kinetic energy, some 300 teratons TNT equivalent, produces a stupendous explosion that melts, vaporizes, and ejects a substantial volume of calcite, granite, and water. The dominant feature in this picture is the curtain of the debris that has been ejected and is now falling back to Earth. The ejecta follows ballistic trajectories with its leading edge forming a conical surface that moves outward from the crater as the debris falls to form the ejecta blanket. The turbulent material interior to the debris curtain is still being accelerated upward by the explosion produced during the excavation of the crater.

For this discussion, Gisler's 3D simulations started on the White machine and continued on to the ASCI Q machine at Los Alamos. The scale of these runs is comparable to those of the milepost runs and the highly-resolved 3D RMI work shown previously. Here we show some images from Gisler's Asteroid impact work: Fig. 11, 12 and 13.

4 Verification and Validation for SAGE and RAGE

During the course of the Crestone Project activities, substantial time has been devoted, either directly or indirectly, to the verification and validation (V&V) of the Project codes. Direct V&V results from a Project team member running specific simulations to compare to analytic results (verification) or to results from experimental data (validation). Indirect validation results from the use of the Project codes by end-users for their own purposes, either in

designing a new AGEX experiment, or in the analysis of a previous experiment, or merely as a tool for analysis of a physical problem. There exists a large suite of verification test problems in our community, including the Sedov blast wave, various geometries for the Noh problem, Marshak radiation wave problems, and many more. Details of the specifications and results from

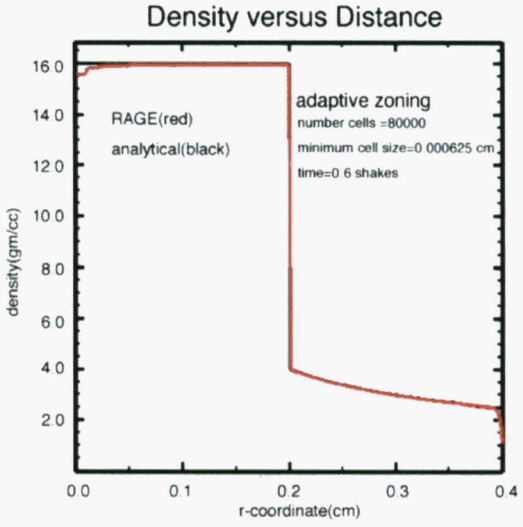

Fig. 14. An example from our verification suite: the cylindrical Noh problem. The red curve is the SAGE CAMR result compared to the black analytic solution.

these types of problems is written up elsewhere and will not be discussed at length here. A detailed report on the suite of verification problems is being prepared (see e.g. – [31],[32]). One example of such a verification problem is the Noh problem, with uniformly converging flow velocities. The results from the cylindrical Noh problem are shown in Fig. 14, comparing the SAGE adaptive mesh hydrodynamics to the analytic result. In general we are very pleased with the success of the SAGE and RAGE algorithms in comparing to analytic and semi-analytic results (see e.g. – [2] for an example of comparisons to semi-analytic problems).

Direct project validation runs are subject to conflict-of-interest discussions, so in this brief section on validation of the Project codes, we will refer the reader to several end-user applications that clearly demonstrate a high level of code algorithm validation (in some cases, e.g. [24], [30] results of code simulations are used for design and prediction of future experiments). The first validation paper published in a refereed journal was "The Simulation of Shock Generated Instabilities" [4] in which detailed code simulations of shock tube experiments were quantitatively compared to the experimental results. The results from this work are summarized in Fig. 15.

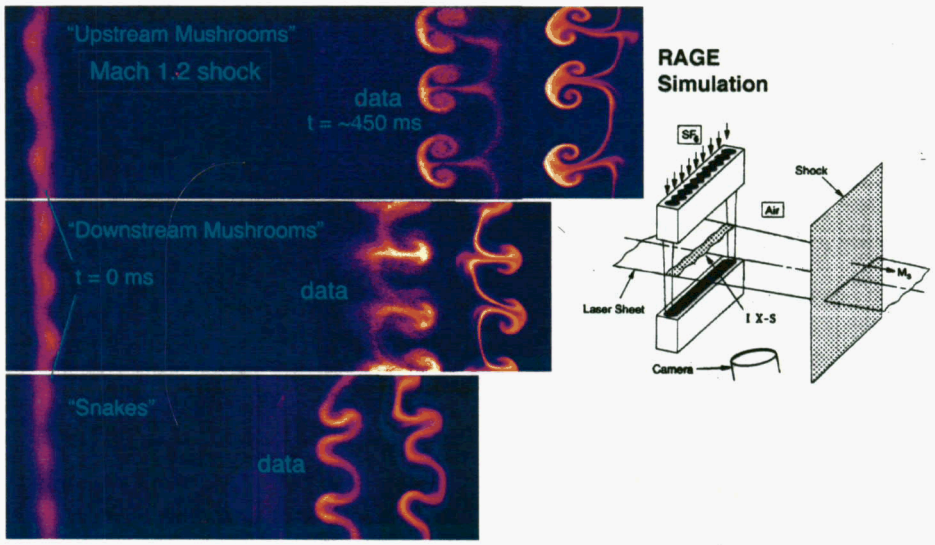

Fig. 15. A graphical comparison of three shock tube experiments to the corresponding RAGE simulations. The three experiments had different initial conditions for the SF_6 gas curtain in the shock tube that resulted in three drastically different dynamical instabilities. The RAGE CAMR hydrodynamics solutions for these experiments is shown offset from the data and is analyzed quantitatively in [4].

Another example of code validation is given in C. Zoldi's Ph.D. thesis for SUNY Stony Brook [33], [7]. In this work, code simulations were done for a similar shock tube environment but in this case the mach 1.2 shock interacted with a dense cylinder of gas, instead of a gas curtain. The reader is referred to her thesis work, with one example of comparisons between experimental images and SAGE results shown here in Fig. 16.

The final example here for validation is the work being done as a multi-lab, indeed, a multinational collaboration on the simulation of supersonic jets and shock interactions [24]. I refer to this work as validation of the "complex hydrodynamics" of the CAMR Eulerian Godunov algorithms in RAGE. Detailed, quantitative inter-comparisons have been done among several computational tools (including RAGE) and the actual experimental data obtained from a series of AGEX experiments. All the code techniques, including CAMR RAGE and ALE codes from both Livermore and AWE (the UK Atomic Weapons Establishment), show good agreement with the data.

These examples touch on the breadth of validation efforts completed and underway in the on-going effort to understand the range of capability for the CAMR Eulerian-based hydrodynamics and radiation energy flow as coded in the massively-parallel, modular version of the RAGE code. More detailed documentation efforts are underway. Two of the areas of active research in the Crestone Project currently are the treatment of interfaces between materials

Fig. 16. A graphical comparison of shock tube experimental images to the corresponding SAGE simulations. The images show a progression of dynamical shapes as the evolution of the initial cylinder moves downstream. Also included in Zoldi's work is a good comparison of code results for velocity distributions to those obtained from the experiment.

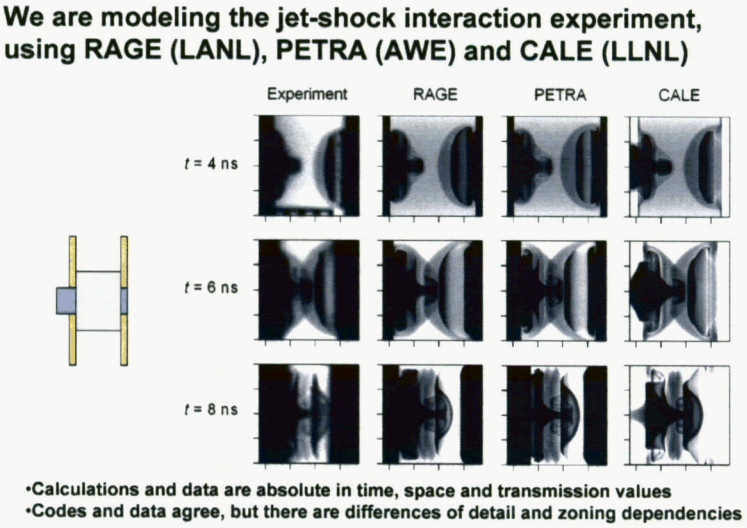

Fig. 17. Multinational modeling of complex hydrodynamics from supersonic jets and shock interactions compared to experimental results. RAGE code was used in the design of the experiments.

and the improvement of the material strength package. Both of these areas of physics in the project have been flagged for improvement and significant progress in this direction is forthcoming.

5 Summary

An overview of the Los Alamos Crestone Project has been given. This project is arguably one of the key success stories of the DOE ASCI program. We have shown a progression of capabilities in massively-parallel simulations, starting from runs of order 100 processors to state-of-the-art multi-physics simulations using thousands of processors and consuming months of supercomputer time. We have demonstrated that the RAGE code is a sophisticated CAMR hydrocode, with a long history of verification and validation. The Crestone Project user community tends to dominate the cycles used on all available high-end supercomputers, but the codes SAGE and RAGE can also be run efficiently on desktop machines, even Macintosh's under MacOS X unix environment. The Crestone Project user-community is fully utilizing each new ASCI supercomputer that is delivered to the complex.

We hope to have demonstrated from the discussion here that massively-parallel computing is fundamentally changing the way we think about computer simulations: simulations that even just a few short years ago were *unthinkable* are not not only being done, but are becoming routine. The 2D and 3D simulations performed by the Crestone Project team and its end-user community in the last two years represent orders of magnitude more computation than was done by all DOE calculations previously performed.

Acknowledgement. The authors would like to thank all the members of the Crestone Project team, both past and present. Special thanks go to Mike Clover and Bill Archer, the former and current code-project leaders. It takes a very special talent to coordinate and actually develop and improve such complex, multi-physics integrated codes as exist in the Crestone Project. We have been extremely fortunate to have found that talent in these two individuals. The unique collaboration between SAIC and Los Alamos has proven to be highly successful, primarily due to the outstanding individuals on our team. Bob Greene and Bob Kares continue to produce outstanding graphics from our simulations using the Ensight visualization software (which is a product of CEI, Inc.) many of which are shown in this paper.

This work was supported by the University of California, operator of the Los Alamos National Laboratory under Contract No. W-7405-ENG-36 with the U.S. Department of Energy.

References

1. "The SAGE Code Multimaterial Equation of State Methods" M.L. Gittings in Numerical Methods Symposium, 28-30 April 1992; "The RAGE Code" R.N.

Byrne, T. Betlach, and M.L. Gittings in Numerical Methods Symposium, 28-30 April 1992. Copies may be ordered from the Defense Nuclear Agency (currently the Defense Threat Reduction Agency), 56801 Telegraph Road, Alexandria, VA 22310-3398.

2. "2D and 3D simulations of RM instability growth with RAGE: a continuous adaptive mesh refinement code" R. Weaver, M.L. Gittings, R.M. Baltrusaitis, Q. Zhang and S.Sohn, July 1997, proceedings of the 21st International Symposium on Shock Waves (ISSW21), Australia, paper 8271.
3. "The parallel implementation of RAGE: a 3D continuous adaptive mesh refinement radiation-hydrodynamics code" R. Weaver, M.L. Gittings, M.L. Clover, July 1999, proceedings of the 22st International Symposium on Shock Waves (ISSW22), London, paper 3560.
4. "The Simulation of Shock-Generated Instabilities" R.M. Baltrusaitis, M.L. Gittings, R. Weaver, R. Benjamin and J. Budzinski 1996, Physics of Fluids, 8 (9), p. 2471 (also as LA-UR-95-4000).
5. "Predictive Performance and Scalability Modeling of a Large-Scale Application" Darren J. Kerbyson, Hank J. Alme, Adolfy Hoisie, Fabrizio Petrini, Harvey J. Wasserman, and Michael Gittings, in Proc. of IEEE/ACM SC2001, Denver, November 2001.
6. "Verifying Large-Scale System Performance During Installation using Modeling" Darren J. Kerbyson, Adolfy Hoisie, and Harvey J. Wasserman, in Hardware/Software Support for Parallel and Distributed Scientific and Engineering Computing, L.T. Yang (eds), Kluwer, September 2003.
7. "Code Validation Experiments — A Key to Predictive Science" Brian Fishbine, Los Alamos Research Quarterly, Fall 2002, LALP-02-194.
8. "Two and Three Dimensional Calculations of a NOVA Hydro Instability Experiment" B.H. Wilde, F.J. Swenson, R.P. Weaver, M.L. Gittings, and W.J. Powers Proceedings of the 6th International Workshop on Compressible Turbulent Mixing, France, July, 1997.
9. "Hydrothermal Pressure Instabilities related to Magmatic Steam Injection and Reflected in Long-Period Seismicity" B.A. Chouet, M.M. Morrissey, M.L. Gittings, and R. Weaver, December 1997, EOS, Transactions, American Geophysical Union 1997 Fall meeting, Vol 78, p F764.
10. "ICF Targets with Joints and Gaps" S. R. Goldman, M. D. Wilke, D.C. Wilson, S.E. Caldwell, W. W. Hsing, R. P. Weaver, Bull. Am. Phys. Soc. 41, 1353 (1996)
11. "Inertial-Confinement Fusion at the Los Alamos National Laboratory: Current Trends in International Fusion Research" Erick Lindman, D. Baker, C. Barnes, B. Bauer, J.B. Beck, R. Berggren, B. Bezzerides, P. Bradley, R. E. Chrien, M. Clover, J. Cobble, C. A. Coverdale, M. Cray, N. Delamater, D. DuBois, B. H. Failor, J. C. Fernandez, L. Foreman, R. Gibson, P. Gobby, S. R. Goldman, D. Harris, A. Hauer, J. Hoffer, N. Hoffman, W. W. Hsing, R. Johnson, K. Klare, R. Kopp, W. Krauser, G. Kyrala, G. Magelssen, R. Mason, D. Montgomery, T. J. Murphy, J. Oertel, G. Pollak, H. Rose, K. Schoenberg, D. P. Smitherman, M. S. Sorem, F. Swenson, D. Tubbs, W. Varnum, H. Vu, J. Wallace, R. Watt, R. Weaver, B. Wilde, M. Wilke, D. Wilson, W. M. Wood, Review and Assessment, Washington, DC, Proceedings of the 2nd Symposium, Plenum Press (March 10-14, 1997).
12. "Two and Three Dimensional RAGE Calculations of a NOVA Instability Experiment" B.H. Wilde, F.J. Swenson, R.P. Weaver, M.L. Gittings, and W.J.

Powers, presented at the JOWOG 32m, Lawrence Livermore National Laboratory, August 13, 1997.
13. "Two and Three Dimensional Calculations of a NOVA Hydro Instability Experiment" B.H. Wilde, F.J. Swenson, R.P. Weaver, M.L. Gittings, and W.J. Powers proceedings of the 6th International Workshop on Compressible Turbulent Mixing, France, July, 1997.
14. "Shock Propagation in the Nonlinear Regime" W.W. Hsing, S. Caldwell, S.R. Goldman, R.P. Weaver, Bull Am. Phys Soc. 42, 1952 (1997)
15. "Shock Structuring Due to Fabrication Joints in ICF Targets" S.R. Goldman, S.E. Caldwell, G.A. Kyrala, M.D. Wilke, D.C. Wilson, C.W. Barnes, W.W. Hsing, N.D. Delamater, J. Grove, J.M. Wallace, R.P. Weaver, A.M. Dunne, M.J. Edwards, P. Graham, B.R. Thomas, Bull Am Phys. Soc. 43, 1667 (1998)
16. "Simulations of Richtmyer-Meshkov Instability in Two and Three Dimensions" Richard L. Holmes, Cindy A. Zoldi, Robert P. Weaver, Michael Gittings and Michael Clover, July 1999, proceedings of the 22st International Symposium on Shock Waves (ISSW22), London, paper 4480.
17. "Shock Structuring Due to Fabrication Joints in Targets" S.R. Goldman, S.E. Caldwell, M.D. Wilke, D.C. Wilson, C.W. Barnes, W.W. Hsing, N.D. Delamater, G.T.Schappert, J. W. Grove, J.M. Wallace, Robert P. Weaver, A.M. Dunne, M.J. Edwards, P. Graham, and B. R. Thomas, Physics of Plasmas, 6(8), pp. 3327 – 3336, 1999. [also as LA-UR-98-4250].
18. "Richtmyer-Meshkov Instability Growth: Experiment, Simulation and Theory" Richard L. Holmes, Guy Dimonte, Bruce Fryxell, Michael L. Gittings, John W. Grove, Marilyn Schneider, David H. Sharp, Alexander L. Velikovich, Robert P. Weaver, and Qiang Zhang, Journal of Fluid Mechanics 39, pp. 55 –79. [Also as LA-UR-97-2606].
19. "Planar,ablative Rayleigh Taylor Instability Growth in Copper Foils" G. T. Schappert, W.W. Hsing,D. E. Hollowell, R. P. Weaver and B. A. Remington 38th Annual Meeting of the Division of Plasma Physics; 11-15 November, 1996; Denver CO
20. "Rayleigh-Taylor Instability Growth in Copper Foils Driven by NOVA Hohlraum Radiation" G. T Schappert, W.W. Hsing,D. E. Hollowell, R. P. Weaver and B. A. Remington; 37th Annual Mtg. of the Division of Plasma Physics; November 6-10, 1995; Louiville,Ky
21. "Ablative Rayleigh-Taylor Instability Modeling" D. Hollowell, G. Schappert, S.Caldwell, W. Hsing, R. Weaver; June 1997 Anomolous Absorption Conference, Vancouver Canada
22. "2-D Rayleigh-Taylor Experiments and Modeling" David Hollowell, Gottfried Schappert, Robert Weaver, Warren Hsing, Rodney Mason, Steve Caldwell; LA UR 98-1828 and 25th European Conference on Laser Interaction With Matter, Formia, Italy May, 1998
23. "Production of Enhanced Pressure Regions due to Inhomogeneities in Interial Confinement Fusion," S.R. Goldman, C.W. Barnes, S.E. Caldwell, D.C. Wilson, S.H. Batha, J. W. Grove, M.L. Gittings, W.W. Hsing, R.J. Kares, K.A. Klare, G.A. Kyrala, R.W. Margevicius, R. P. Weaver, M.D. Wilke, A.M. Dunne, M.J. Edwards, P. Graham, and B. R. Thomas, Physics of Plasmas, 7(5), pp. 2007 – 2013, 2000.
24. "Supersonic Jet and Shock Interactions," J.M. Foster, B.H. Wilde, P.A. Rosen, T.S. Perry, M. Fell, M.J. Edwards, B.F. Lasinski, R.E. Turner and M.L. Gittings, Physics of Plasmas, 9(5) Part 2, pp. 2251 – 2263, 2002.

25. "On 3D, Automated, Self-Contained Grid Generation Within the RAGE CAMR Hydrocode" by W.R. Oakes, P.J. Henning, M.L. Gittings, and R.P. Weaver, in proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, p 973, Chateau Whistler Resort, September 2000.
26. "Two- and Three-Dimensional Simulations of Asteroid Ocean Impacts" by Galen Gisler, Robert Weaver, Charles Mader, and M.L. Gittings, Los Alamos internal report, LA-UR-02-30, 2002.
27. "Two- and Three-Dimensional Simulations of Asteroid Impacts" by Galen Gisler, Robert Weaver, M.L. Gittings, and Charles Mader, Los Alamos internal report,accepted for publication in Computers in Science and Engineering Journal, 2004.
28. "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations" P. Colella and P. R. Woodward, J. of Computational Phys. 54(1):174-201, 1984.
29. "Local adaptive mesh refinement for shock hydrodynamics" Berger, MJ and Collela, P (1989), J. Comp. Phys. 82, 64-84
30. private discussion with Dr. B. H. Wilde of Los Alamos National Laboratory.
31. "2-D Convergence Analysis of the RAGE Hydrocode" J. R. Kamm and W. J. Rider, Los Alamos National laboratory internal report, LA-UR-98-3872.
32. "Comparing the Fidelity of Numerical Simulations of Shock-Generated Instabilities" J. R. Kamm, W. J. Rider, P.V. Vorobieff, P.M. Rightley, R.F. Benjamin, K. P. Prestridge, July 18-23, 1999, proceedings of the 22st International Symposium on Shock Waves (ISSW22), London, paper 4259.
33. "A Numerical and Experimental Study of a Shock-Accelerated Heavy Gas Cylinder" C. Zoldi Ph.D. thesis for the State University of New York at Stony Brook, May 2002.