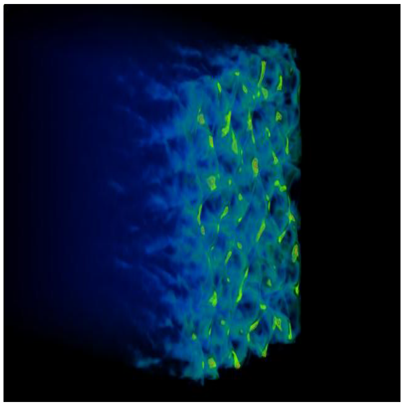
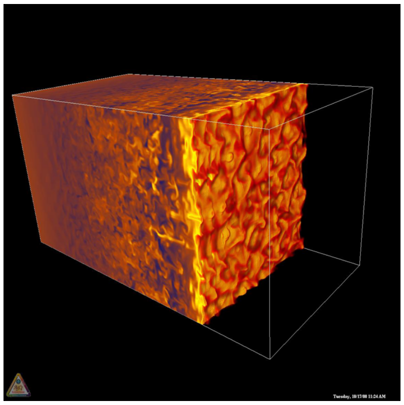
ASCI/ALLIANCES CENTER FOR ASTROPHYSICAL THERMONUCLEAR FLASHES AT THE UNIVERSITY OF CHICAGO YEAR 3 ACTIVITIES REPORT

October 2000





Abstract

We summarize the Year 3 activities at the University of Chicago Center for Astrophysical Thermonuclear Flashes. Major achieved milestones include the refinement of the first production version of the Flash code – a fully-parallelized, adaptive mesh astrophysics code – which has now reached revision level 1.61; completion of a modern architecture version of this production code, *Flash-2.0*; a number of new astrophysics and validation calculations using the production Flash code; performance and scaling studies on all of the ASCI platforms; optimization of existing physics modules, and the development of new physics modules (including modules for self-consistent gravity and magnetohydrodynamics); further investigations of code architectures and advanced code engineering; and a variety of validation, verification, and basic physics studies relevant to the Flash code.

Credits for Title Page Picture: (*Flash-1.61* simulation of a 3-D nuclear cellular detonation under astrophysical conditions (Timmes et al. 2000). The upper panel shows a volume rendering of the pressure field (courtesy ANL), while the lower panel shows the abundance of Silicon ash (courtesy LLNL).

Contents

1 Introduction

1.1 Overview

The "FLASH" problem is centered on simulating the accretion of matter onto a compact star and the subsequent stellar evolution, including nuclear burning either on the surface of the compact star, or in its interior. Our activities involve scientists primarily located at the University of Chicago and Argonne National Laboratory, but also involve a number of collaborators at other universities and at the DOE DP laboratories. Our Center is composed of three disciplinary groups (Astrophysics, Computer Science, and Validation/Basic Science), as well as a cross-cutting group (the Code Group).

In Year 3, we completed the first production version of the *Flash* code (now at revision level 1.61), which is capable of addressing many of our astrophysics problems on the largest existing parallel-architecture computers; we completed the first version of a newly architected version, *Flash-2.0*; we addressed some of the astrophysics issues relevant to X-ray bursts on neutron stars, novae on white dwarfs, and supernovae within white dwarfs; we have made further advances in our understanding of the issues that confront us in massively parallel computing for the very largest problems we want to tackle, ranging from algorithmic issues to data handling and visualization; we have conducted research on how the *Flash* code should evolve, and evolved a definite plan for this future evolution; and we have made further strides in our validation and basic physics program.

1.2 Re-structuring

Following the Year 1 site visit, we re-structured our activities in order to focus our efforts more directly upon building the *Flash* code. This change, which involved the creation of a team of scientists whose specific task during Year 2 was to assemble the first major release of the *Flash* code, has been further streamlined during the past (3rd) year: All of the code building is now carried out within the Flash Code Group, led by Bruce Fryxell. This group has the sole responsibility for building and maintaining the *Flash* code. The Code Group now has an advanced architecture team, led by Andrew Siegel; this team is responsible for defining and constructing new versions of the *Flash* code, and has succeeded in constructing our new architecture version, *Flash-2.0* (described in detail below).

In addition, we have significantly tightened the management structure of the Center. The previous large and unwieldy Advisory Committee was disbanded, and replaced by a small Management Group (composed of the Working Group leaders T. Dupont, B. Fryxell, E. Lusk, and J. Truran, as well as Director R. Rosner and ex officio member R. Stevens), which meets weekly on Friday afternoons.

1.3 Other issues

Both of the previous site visit reports urged us to create two new positions, the first related to a program manager/associate director, and the second related to a code architect. With the completion of our re-structuring, we now have the people infrastructure in place: both the code group as a whole, and the code architecture team, have dedicated leadership. We have demonstrated by direct accomplishment that our re-structuring has worked.

Some concern was also previously expressed about the level of interaction between the Center's activities and computer science. Because of the evolution of the *Flash* code, we have found it natural to strengthen the ties of the Code group to the long-range research effort in architectural and algorithmic issues carried out within the Computer Science Group. Thus, our computer science colleagues were participants in the process which led to the architecting of *Flash-2.0*; and computer scientists are closely involved in research and construction of additional hydro solvers for *Flash*, as detailed in later sections. (Examples of new hydro solvers we are exploring include Discontinuous Galerkin [RPI/ANL] and wENO [UofC], and an anelastic solver for problems in which compressible effects are unimportant other than via gravitational stratification.)

1.4 External advice

In order to obtain a "reality check" on our ongoing efforts, we again asked the chair of the Year 1 site visit team, Prof. Richard Matzner of the Univ. of Texas at Austin, to return during the summer of 2000 to conduct a "mini-site visit". This visit was carried out in August 2000 and complemented a mandated site visit by DOE ASCI scientists (both from DOE Headquarters and from the DP Laboratories) carried out in June 2000.

On the whole, both visits went very well. We were able to show substantial progress on building the *Flash* code; new results for both astrophysics and validation; and our progress in code architecture developments. This report summarizes all of these results, as well as subsequent results obtained in the interim period.

2 The *Flash* Code

Participants: A. Caceres[1], A. Calder, T. Dupont, J. Dursi[1], B. Fryxell (Group Leader), T. Linde, A. Mignone[1], K. Olson, P. Ricker, R. Rosner, K. Riley, A. Siegel (Code architect), F. Timmes, H. Tufo, N. Vladimirova, G. Weirs, K. Young, M. Zingale

During the past year, the Code Group was reorganized to include the former Computational Physics Group. The new combined group is responsible for the

[1] Graduate student

architecture of the code framework, incorporating physics modules developed by other groups, maintaining the code (including code verification), and writing and maintaining the documentation. In addition, members of the Code Group are responsible for constructing new modules for hydrodynamics, MHD, and radiative transfer, as well as improving the adaptive mesh package (PARAMESH). Validation of the code is a joint effort of the Code Group and the Validation & Basic Science Group. Code optimization and scaling studies on the ASCI computers is performed in collaboration with the Computer Science Group. Visualization capabilities have been dramatically enhanced through interaction with the Futures Laboratory at Argonne National Laboratory. Finally, limited support is provided to help new customers in the astrophysics community to begin using the code.

The *Flash* code has developed significantly during the past year. The current production version has evolved from *Flash-1.0* to *Flash-1.61.* The latest version contains several improvements to the code framework, primarily to improve modularity. In addition, a more efficient version of PARAMESH has been implemented, along with a number of new physics modules. A module enabling parallel I/O has also been added, and performance of the code has been enhanced by approximately a factor of two on the ASCI platforms. This version is capable of addressing many of our target astrophysics problems. Construction of *Flash-2.0* has now been completed, and is being subjected to verification tests. This version incorporates a new framework using a more object-oriented approach. This new, more flexible framework makes adding new modules for hydrodynamics, MHD, and radiative transfer much easier and also makes the code easier to maintain, with only a small ($< 10\%$) performance penalty. These efforts will be described in more detail below.

2.1 The physics of *Flash*

The current version of the *Flash* code includes the following physics:

- Compressible hydrodynamics. The current default algorithm is an explicit higher–order Godunov method based on the Piecewise Parabolic Method (PPM) of Colella & Woodward [6], derived in its present form from the PROMETHEUS code [9]. Modules which make use of other algorithms will be described below.

- Arbitrary equations of state. Each problem – from astrophysics to verification or validation – requires its own equation of state. Typically, we use computationally-optimized equations of state based on table lookup and interpolation [40, 42], though in some circumstances far simpler equations of state, such as a gamma law, suffice and are available. New equation of state modules can be added easily when required.

- Arbitrary nuclear reaction network. Any number of nuclear species and reactions can be included up to the memory limits of current computers [38, 41]. The choice of network depends on the initial composition and thermodynamic state of the material and whether we are interested in detailed nucleosynthesis or just need a good approximation to the energy generation rate. For problems not involving nuclear burning, the reaction network module can be turned off.
- Gravity. An external gravitational acceleration can be specified *a priori*, or the gravitational field can be computed self-consistently via a Poisson solver.
- Thermal conduction, in the diffusion approximation; we use explicit time integration, which suffices for the subset of astrophysics problems we have been considering to date.

The physics just described for *Flash* leads to the following set of equations, which govern the motion of compressible matter undergoing nuclear burning in the presence of gravitational stratification: To begin with, we require a continuity equation

$$\frac{\partial \rho}{\partial t} + \bigtriangledown \cdot (\rho \mathbf{v}) = 0, \tag{1}$$

where ρ is the gas density, and $\mathbf{v}$ is the gas velocity. The motion of each nuclear species must be followed independently by solving the set of advection–diffusion equations

$$\frac{\partial \rho X_i}{\partial t} + \bigtriangledown \cdot (\rho X_i \mathbf{v}) = \bigtriangledown \cdot \rho D_i \bigtriangledown X_i + \rho \dot{X}_{i_{nuc}}, \tag{2}$$

where X_i is the mass fraction of the i'th species, D_i is the corresponding diffusion coefficient, and $\dot{X}_{i_{nuc}}$ is the change in composition of the i'th species due to nuclear burning. For most of our target astrophysics calculations, the species diffusion term can be ignored. The equation for conservation of momentum then takes the form

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \bigtriangledown \cdot (\rho \mathbf{v}\mathbf{v}) = -\bigtriangledown P + \bigtriangledown \cdot \sigma - \rho \bigtriangledown \Phi, \tag{3}$$

where P is the gas pressure, σ is the viscous stress tensor, and Φ is the gravitational potential. For Type Ia supernova simulations, it is necessary to compute the self-gravity of the star; in this case, the gravitational potential is obtained by solving Poisson's equation

$$\bigtriangledown^2 \Phi = -4\pi G \rho \tag{4}$$

where G is the gravitational constant. Under more restricted conditions (e.g., studies of the evolution at small spatial scales, or X-ray bursts on a neutron star's surface), stellar expansion can be ignored, and one can assume that the last term in the momentum equation can be replaced by a static spherical gravitational acceleration. *Flash-1.61* and *Flash-2.0* can now deal with any of these cases.

Energy balance is computed by solving the corresponding equation for energy conservation,

$$\frac{\partial \rho E}{\partial t} + \bigtriangledown \cdot (\rho E + P)\mathbf{v} = \bigtriangledown \cdot (\mathbf{v}\sigma - \mathbf{q}) - \rho \mathbf{v} \cdot \bigtriangledown \Phi + \rho \dot{\epsilon}_{nuc}, \tag{5}$$

where

$$E = \epsilon + \frac{1}{2}\mathbf{v}^2 \tag{6}$$

is the sum of the specific internal and kinetic energies and $\dot{\epsilon}_{nuc}$ is the specific rate of energy generation by nuclear burning. Due to the high density of the gas in compact objects, energy transport is entirely in the diffusive regime for much of their temporal evolution (for Type Ia supernovae, this is not true in the ejected envelope); in that case the diffusive energy transport flux, including both radiation and conduction, is given by

$$\mathbf{q} = \frac{-4acT^3}{3\rho}\left(\frac{1}{\kappa_{rad}} + \frac{1}{\kappa_{cond}}\right) \bigtriangledown T, \tag{7}$$

where a is the radiation constant, c is the speed of light, T is the temperature, κ_{rad} is the radiative conductivity, and κ_{cond} is the (electron) thermal conductivity [39]. The last term in the energy equations represents the heat generated by nuclear burning. At this point, the energy equation adopted by *Flash*, versions 1.0-1.61, assumes that radiation and conduction operate fully in the diffusive regime and is solved fully explicitly; these approximations suffice in cases in which the diffusion time scale is longer than the CFL time. Finally, the equations are closed by an equation of state

$$P = f(\rho, \epsilon) \tag{8}$$

which consists of a mixture of electron degeneracy pressure, radiation pressure, and ideal gas pressure. These are the equations that *Flash* is designed to solve; an overview of additional physics that we plan to incorporate into the *Flash* Code is discussed below.

2.2 *Flash* Code Structure

The current version of the *Flash* code [10] represents a major advance along the road to the ultimate goal of a fully flexible code for solving general astrophysical fluid dynamics problems. *Flash-1.61* solves the equations described above, is modular and adaptive, and operates in parallel computing environments. It has been designed to allow users to configure initial and boundary conditions, change algorithms, and add new physical effects within certain limits. It uses PARAMESH [28] to manage a block-structured adaptive grid, placing resolution elements only where they are needed most. Inter-processor communication is accomplished using the Message-Passing Interface (MPI) library to achieve portability and scalability on a variety of different message-passing parallel computers [13]. To date, it has been successfully tested on a variety of Unix-based platforms, including

- SGI systems, *e.g.,* the Nirvana Cluster at LANL
- SP-2 at ANL, ASCI Blue Pacific and ASCI White at LLNL, and Blue Horizon at UCSD, all built by IBM
- ASCI Red at SNL, built by Intel
- Intel-based systems running Linux ("Beowulf" systems), such as Chiba City at ANL, and the Hive cluster at NASA/Goddard Space Flight Center
- Alpha-based clusters, such as CPlant at SNL
- SGI/Cray T3E (at PSC/Pittsburgh)

2.2.1 *Flash-1.x*

Flash-1.x has passed through several revisions during the past year. Version 1.5 was released to the Center on November 23, 1999 and featured several framework modifications which improved the modularity of the nuclear reaction networks and directionally split hydrodynamics routines. However, extensive use of global variables still led to overly tight coupling among components, and many basic simulation framework services were still lacking.

Version 1.6 of the *Flash* code was released for general use by the Center on March 27, 2000. Following an intense initial debugging period, it became the production version of the code. *Flash-1.6* introduced a number of significant improvements to the *Flash* code framework. In particular, it was the first version of the framework that was consciously designed to mimic an object-oriented hierarchy. Since the F90 language itself does not directly support class inheritance and polymorphism, code "glue" was developed to essentially achieve the same effect at compile time – i.e., a script to piece together the user-selected implementations of the *Flash* "abstract methods" or "virtual function". Furthermore, accessor and mutator methods were used with F90 modules to achieve superior data protection and encapsulation. Many (but not all) cross-module included common blocks were eliminated (complete elimination is the goal of *Flash-2*, described below); all physics modules accessed only their own data and the

PARAMESH data structures through included commons, and communicated with each other only through the new accessor functions.

Version 1.6 also implemented several basic simulation framework services. These included a message logging facility; a performance monitoring library that supplies a set of named clocks for timing segments of code; a physical constants database capable of performing arbitrary unit conversions; a runtime parameter database that supports multiple parameter contexts (allowing modules to encapsulate their runtime parameters); and a materials module, which collects together information on fluid properties and equations of state and makes this information available to the rest of the code through accessor functions.

A steady-state driver module has been added to enable calls to physics modules which do not require time evolution. This would be used, for example, for a single call to a Poisson solver module.

During July extensive changes were made to improve performance, yielding the current production version, *Flash-1.61*. Release of the code to the astrophysics community took place in October 2000.

2.2.2 *Flash-2.0*

Flash-2.0 is built around a number of significant core architectural changes aimed at simplifying the development, maintenance, and re-usability of the *Flash* framework. These changes are aimed both at application developers (users who wish to customize the code by adding their own physics, numerical strategies, etc.) and our own in-house developers, who benefit greatly from a more modular design. Additionally, new tools have been added to simplify the experience of the end-user who is interested only in running the current form of the code.

The *Flash-2.0* architecture makes a clear distinction between the *Flash* "framework", which defines algorithmic interfaces and the main thread of execution, and the particular physics modules. As is typical in modern software architecture, the framework controls the thread of execution and makes calls to various abstract methods, which can easily be interchanged if they adhere to a common interface. When a single common interface is difficult to identify, various Design Patterns can be employed to allow some flexibility across different implementations and retain plug-and-play capabilities.

An important decision in developing *Flash-2.0* centered around choosing a language for the framework. The physics modules themselves should always be language-independent, requiring only the proper inter-language bindings on the platform in question. However, the choice of framework language brought a number difficult issues to the forefront. Java, for example, has excellent support for object-oriented design concepts, but is very weak on performance, interoperability with FORTRAN, and usability with MPI (no official Java bindings to MPI have as of yet been defined). C++ offers good performance but is notoriously difficult to port. Furthermore, a certain degree of sophistication in C++

design, uncommon among scientific programmers, is required to avoid programming nightmares. Also, given the breadth and sophistication of *Flash-1.6*, we strongly favored a strategy that allowed incremental testing and backward compatibility with *Flash-1.6*. After weighing these and other issues, we ultimately chose to implement the architecture using Fortran 90 (F90).

After defining an incremental testing strategy, the first step was to eliminate all common block data. A "database" was then developed as an F90 module that warehoused all of the previously common grid data and parameters that were not specific to an individual subroutine (the F90 module type mimics a singleton class in C++, and behaves similar to a class in Java with all static variables). The database is then the mechanism by which the framework shares data – each framework subroutine, program, or module contains a reference to the database and accesses its private variables through accessor and mutator methods. The database contains a rich set of overloaded methods which hide the details of re-packaging the data for different modules (getDataXSlice, getDataTranspose, etc). Furthermore, F90 intent statements are used to clarify the purpose of each variable.

The physics modules themselves communicate only through interfaces and may not access the database variables. This ensures that the physics modules know nothing of the framework in which they exist and facilitates the swapping of new modules or the incorporation of new framework services (such as an alternate AMR package). Our solution to the common interface problem was initially a simple one – to pass the maximum amount of data that could reasonably be expected to be needed by the corresponding physics module and to have each implementation choose what subset it needed. More sophisticated approaches are being considered for future versions of *Flash*.

In choosing this approach, we retained the essentially polymorphic structure of *Flash-1.6* by using a pre-compile-time setup script to glue together the proper physics modules required for a specified application. However, since the setup script is such a complicated and important part of the code, we chose to implement it entirely in the Python language, rather than as a mixture of cshell, awk, and sed. This implementation has greatly increased the ability of the script to grow easily in proportion to the increasing complexity of the *Flash* component hierarchy. Furthermore, by incorporating GNU's autoconf into the setup, we have improved the portability of the code onto clusters on which it has never been tested.

Finally, significant progress was made both in the automated testing and usability of the code. For the former, a Python-based testing utility (*FlashTest*) was developed to allow automated job submission, ouput comparison, and fidelity testing. Regarding the latter, a Globus-based Java-Swig front end was developed that facilitates the setup and deployment of the code across a client-server architecture.

2.3 Adaptive Mesh Refinement

Flash achieves reduced time to solution and improved accuracy and efficiency through the use of Adaptive Mesh Refinement (AMR). Adaptivity is managed by PARAMESH [28], a block-structured AMR package developed at NASA/Goddard Space Flight Center. A number of very important improvements to PARAMESH have been made during the past year, which significantly enhance the capabilities of the *Flash* Code.

2.3.1 PARAMESH-1

The original version of PARAMESH was written for the Cray T3E and took advantage of the 'SHMEM' library, which allowed one processor to directly read and write data to the memory of another processor. For machines which did not support 'SHMEM', one-sided communication fuctions were simulated using the MMPI library. This approach required the use of many synchronization barriers throughout the code and achieved poor performance on the ASCI platforms. It also resulted in a code which was extremely difficult to debug. During the past year, the MMPI library was completely removed from PARAMESH and replaced with a native MPI implementation, and all extraneous barriers were removed.

2.3.2 PARAMESH-2

Over the past year progress has also been made in developing a completely new version of PARAMESH, which can run in either MPI mode or in 'SHMEM' mode. This new version extends the capabilities of the old version in several ways. The data structure in PARAMESH is composed of blocks of cells which are each logically Cartesian; these blocks are layed out to completely cover the computational domain and can have different resolutions at different places in the domain. Each block must exchange data with any neighboring blocks it might have. The old version of PARAMESH handled this by allocating a layer of permanent guard cell storage around each block. For the *Flash* code, this guard cell layer is four cells wide and accounts for a large memory overhead. The new version of PARAMESH no longer requires this permanent guard cell storage. Other new features of PARAMESH include the ability to advance the solution at all levels in the mesh hierarchy, to advance the solution at different time steps at different places in the mesh, and to fill guard cell data in only one particular coordinate direction at a time. PARAMESH-2 is being tested with *Flash-1.61* and will be fully implemented into *Flash-2.0*.

Finally, the AUTOPACK library developed by R. Loy of the FLASH Center provides automatic message packing and asynchronous reduction operations for high communications efficiency with a minimum of effort by the application programmer. We are exploring the incorporation of AUTOPACK into the next version of the PARAMESH library.

2.4 Physics Modules

The *Flash-2.0* framework was specifically designed to accomodate several new physics modules that are currently under development. In addition, some of the original modules have been upgraded with new capabilities.

2.4.1 Hydrodynamics

In conjunction with the testing of *Flash-2.0*, work on several alternative numerical techniques for treating the Euler terms and solution advancement has intensified. Several Runge-Kutta modules are being developed for time advancement. In addition to the PPM method currently employed, TVD, weighted ENO, and high-resolution central shock-capturing schemes will be available, as well as non-dissipative central differences.

Alternative time advancement methods require a "delta-formulation", which would require myriad changes throughout *Flash-1.x*. Implementing the delta formulation in *Flash-2.0*, while not trivial, is much more tractable and less likely to interfere with the sections of the code already debugged, tested, and optimized.

We are continuing research on numerical methods for hydrodynamics. This involves developing new schemes, testing recently published techniques, and performing further analysis on the methods we have; details and specific examples are provided in §2.8 below.

2.4.2 Magnetohydrodynamics

The *Flash* MHD module currently consists of two sets of routines that enable the end user to solve ideal MHD problems as well as viscous, resistive problems with heat transfer. The latter routines are implemented using second- and fourth-order explicit finite difference schemes of central type. This way the users have an option of running fast but relatively low-resolution submodules when they setup their problem and then using high-resolution but somewhat slower submodules for production runs. The ideal MHD routines are implemented using the MUSCL TVD method [48] with an option of choosing from a set of the Lax-Friedrichs, HLLE or Roe interface flux functions. The divergence of the magnetic field is controlled using the algorithm of Powell [30] combined with a monopole diffusion method [26]. Our numerical experiments show that this technique can successfully suppress the generation of magnetic monopoles. We also retain the option of using the elliptic projection method [1]. The *Flash* MHD module has been extensively tested on a number of model problems and is currently being incorporated into *Flash-2.0*.

2.4.3 Self-Gravity

We have implemented three Poisson solvers in *Flash-1.61*: a fast multipole solver, a multigrid solver, and an adaptive FFT solver. The multipole solver directly sums fields due to the multipole moments of the matter distribution within the simulation volume, up to some limiting multipole order $\ell_{\max}$. This is particularly efficient for nearly-spherical, isolated situations like the supernova problem. The multigrid solver is appropriate for more general mass distributions. It makes efficient use of the entire PARAMESH block hierarchy, allowing solutions on irregularly refined meshes using periodic or isolated boundaries. The latter case is handled using an image mass technique [14]. Finally, the adaptive FFT solver, based on the particle-particle-particle-mesh (P^3M) algorithm [8], is designed to efficiently compute the potential due to a collection of particles on an adaptive mesh. We are investigating this method as a complement to the first two methods for future collisionless particle simulations (e.g., shock acceleration of cosmic rays), although our current implementation uses grid-based density data rather than particles.

2.4.4 Radiation Transport

A module for single-group flux-limited diffusion on an AMR grid is currently under development. Extensions to multi-group transport will then be implemented. Initially, we will assume that the radiation is tightly coupled to the matter (one temperature). Substantial progress has already been made in the development of an implicit diffusion equation solver, which allows for the solution of the linear or non-linear diffusion equation on the PARAMESH grid structure using a V-cycle multigrid method. Several differencing schemes have been applied to this problem, including Crank-Nicholson and backward Euler differencing. The method has been tested on simple diffusion problems, and is currently being applied to standard radiation transport test problems.

2.4.5 Equation of State and Nuclear Burning

Coulomb corrections have been added to the equation of state for partially degenerate and relativistic stellar matter in a thermodynamically consistent way. A new gamma-law equation of state module has been added which permits using multiple fluids, each with a different value of gamma. In addition, the nuclear burning module has been upgraded to permit computing reaction rates from a table lookup instead of from complex analytic formulae. The performance improvement depends on which network is being used but is typically a factor of two.

2.5 Parallel I/O

The I/O section of the *Flash* code has been upgraded to support HDF-5 output, which is a more flexible file format than its predecessor. The major advantages of this format are support for files larger than 2 GB and parallel I/O through ROMIO. We have implemented both a serial and a parallel HDF-5 module in *Flash* and have tested them on the ASCI computers. Without parallel I/O, our recent production calculations (see §3) would have been impossible to complete.

Although the parallel I/O is faster than the serial version we previously used, the sustained bandwidth is only a small fraction of the peak performance on the ASCI machines. The best performance has been on Blue Pacific. The most likely reason for the poor performance is that our application writes only a small fraction of the memory on each processor to disk. Throughput is a strong function of the amount of data being written at once. We have recently begun collaborating with the HDF development team at NCSA and with the parallel I/O group at LLNL and have provided both groups with an I/O benchmark that replicates the I/O performance of *Flash*. They are investigating ways to improve both the *Flash* code and the performance of the HDF 5 library for small record sizes.

2.6 Performance and Scaling

The *Flash* code underwent a large optimization effort in preparation for this year's integrated calculations. The performance of *Flash-1.61* is now more than twice that of *Flash-1.6*. These optimizations included both single processor tuning and parallel performance improvements.

Single processor performance tuning made extensive use of the profiling tools on the different platforms – primarily speedshop/perfex on the SGI, and compiler-based profiling on ASCI Red. Our optimizations included reducing the number of divides and square roots, consolidating and eliminating scratch arrays, and eliminating unnecessary array copies and initialization. String comparisons were replaced by integer comparisons for the various databases maintained by *Flash*. Calls to math library functions were optimized by using the vendor supplied libraries (-lmass on Blue Pacific, and -lfastm on the SGIs), loops were fused together to eliminate unnecessary storage, Cray vector merge routines were removed from legacy code, and changes were made to often-used routines to allow ASCI Red's compiler to inline them. *Flash-1.61* now achieves greater than 90 Mflop/s with 64-bit arithmetic on a single 250 MHz R10000 processor for simulations that exercise all of the major modules of the code.

Parallel performance was improved greatly on Nirvana and Blue Pacific through the elimination of unnecessary barriers in PARAMESH; the Jumpshot performance visualization tool, being developed as part of the FLASH computer science effort, partly in collaboration with IBM and LLNL, has been useful in this effort. A notable accomplishment was the removal of the large number of barriers in the PARAMESH library which had been inherited from the T3E version of the library when it was converted to MPI. ASCI Red, which has better support for collective operations such as barriers and allreduces, did not suffer as greatly from the barriers in the code. Some portions of the code benefited from explicit message packing – grouping smaller messages together before communicating across processors. There are a large number of small messages in the tree portions of the code; performance tools such as Vampir (on Blue Pacific) and Jumpshot (developed at ANL) helped find these problems and suggest solutions. Permanent 'hooks' were added to the code to make monitoring with these packages straightforward.

Scaling of the *Flash* Code has been tested on a wide variety of computers, including all three of the ASCI platforms. Figure 1 shows one such study. The simulation used for this test was a three-dimensional cellular detonation (see §§3-4). This problem was chosen becaise it exercises most of the major physics modules in the code (adaptive mesh, hydrodynamics, eos, and burning). Our approach was to pick a fixed problem size and begin with the smallest number of processors on which the calculation would fit. The number of processors was then increased until there was too little work on each one to expect good scaling. At this point, the problem size was increased and the study continued to larger numbers of processors.

The smallest problem sizes used 5 levels of refinement, with 1380 8^3 blocks at the beginning of the calculation and 2060 blocks at the end. The number of blocks in the largest (7 refinement level) calculation ranged from 21869 to 32780. The plots clearly show a modest deviation from ideal scaling. This is primarily due to the fact that the fraction of blocks which need to get their guard cell information from blocks on other processors increases as the number of blocks per processor decreases. Thus, the ratio of communication to computation increases with processor number. In practice, we run the code on the smallest number of processors on which the problem will fit, so we are always operating at peak efficiency. We will never reach the point where the problem size we want to run is too small for the number of processors available.

The *Flash* Code recently achieved 0.25 Tflops on 6420 processors of ASCI Red. A paper describing the optimizations and performance of the code [3] has been selected as a finalist for the Gordon Bell prize at Supercomputing 2000.

2.7 Documentation

Details regarding the *Flash* code can be found at our Center web site,

- http://flash.uchicago.edu/flashcode

The documentation includes a detailed user's manual, which can be found at

- http://flash.uchicago.edu/flashcode/doc.

A scientific paper [10] describing the physics and algorithms in the *Flash-1.0* can be found at

- http://flash.uchicago.edu/flashcode/pubs

Finally, a FAQ based on user feedback has been started, and will evolve with our growing user base. This document is currently distributed in the FLASH directory structure. Copies of the documentation will be made available at the Site Visit.

2.8 Future developments of the *Flash* code

Evolution of *Flash* from version 1.0 to version 1.61, was the first major step in our code development. The next steps in its development relate to refinement of its code architecture. *Flash-2.0* will continue to evolve to incorporate new object-oriented features. Patch-based AMR packages, such as AMRA [29], will be explored as possible alternatives to PARAMESH. The modularity of the code will be enhanced to permit addition of new physics modules which are not supported by the current framework. Some of these new modules are listed below.

1. Relativistic hydrodynamics. For some of our target astrophysics problems, special relativistic effects can become important. It is therefore important to have at least the capability to test for the consequences of such effects. We have developed a special relativistic version of the PPM hydro module, which has been tested on a relativistic jet problem, and will be incorporated into *Flash-2.0*.

2. Discontinuous Galerkin techniques. For parallel codes, communication overhead can be reduced by using algorithms with a very compact stencil for spatial difference operators. We are conducting a comparative study of one such family of techniques, namely Discontinuous Galerkin (DG) methods using a wide variety of analytic test problems. This study also includes a comparison of DG results on structured meshes with those obtained with *Flash-1* for Rayleigh-Taylor instabilities.

3. Subsonic hydrodynamic solvers. During the early phases of our target astrophysics simulations, fluid motions are very subsonic. In some cases, gravitational stratification may in addition be weak. An example is convection near or at the center of an evolved star. In such cases, one saves considerable computational effort by filtering out sound waves (the anelastic approximation) and, if permissible, additionally ignoring gravitational stratification (leading to the Boussinesq approximation). We are pursuing two complementary avenues to address these simplifications: first, we plan to implement an anelastic spectral element hydro solver within the *Flash-2.0* framework. Second, we have developed a semi-implicit compressible

hydrodynamics module [33]. Fully implicit hydrodynamic algorithms are also begin considered.

2.9 *Flash* code verification

Code verification – as opposed to validation – focuses on answering the questions: Are there bugs or outright errors in the codes? Are the computed solutions suitably close to desired solutions of the mathematical model? The latter question involves both convergence (are the solutions to the discretized equations appropriate approximations to the solutions of the adopted model equations?) and a posteriori error estimation (are the solutions accurate, i.e., are the computational errors suitably bounded?).

While it is not trivial to assert with any certainty that a given, highly complex, hydrodynamic code is free from bugs and outright errors, one can devise procedures to guard against such difficulties. We have constructed a suite of test problems, which include the Sod shock-tube problem [36], the Sedov explosion problem [35], the Woodward-Colella two-blast-wave problem [51], an advection problem in which we create a planar density pulse in a region of uniform pressure; a double Mach reflection problem; and a wind tunnel flow with a step. Several new test verification test problems have been added to *Flash-1.6* including a stand-alone burn test problem, a Jeans instability problem, the SAMRAI explosion-in-a-room test problem, the Shu-Osher problem, and a self-similar spherical collapse problem. As the *Flash* code gains new capabilities, additional problems will be added to the suite to test each module individually and in combination with other modules. Our procedure is to test each new version of the *Flash* code against these test problems before this version is committed to our (CVS) code repository; an automatic procedure for such testing is part of the *Flash* programming environment.

2.10 *Flash* code validation

We have now identified a number of experimentally well-studied problems which can serve both as validation problems for the *Flash* code and as "laboratories" for improving our understanding of physical processes relevant to our astrophysical applications. These problems include hydrodynamic and magnetohydrodynamic Rayleigh-Taylor, Richtmyer-Meshkov, and Kelvin-Helmholtz instabilities; laminar and turbulent flame front propagation; interface dynamics; multiply-diffusive systems; and interface waves in conducting fluids. Our progress in these various areas is summarized in §§6-7.

2.11 *Flash* code visualization

A major challenge for the practical use of the *Flash* code is effective visualization of results obtained by this code. Because of the enormous size of the data sets, including data sets whose spatial dynamic range exceeds what can be displayed on a standard workstation screen or whose spatial complexity is sufficiently large that it cannot be readily appreciated by taking 2-D slices of the data, the Center's Computer Science visualization effort has indeed been challenged. Our early results from using both the CAVE and the ActiveMural, shown at last year's site visit, suggest that such advanced visualization techniques are extremely useful. This past year, substantial progress has been made in a number of areas, including volume rendering, using the native PARAMESH data format rather than interpolating onto a uniform mesh, and so forth. These efforts are detailed in §5 below.

3 The Integrated Calculation: A Nuclear Cellular Detonation

3.1 Background

A hallmark of virtually all astrophysical calculations is that it is not possible – even in our wildest dreams – to simulate directly all of the relevant spatial and temporal scales of the astrophysical problems of interest. The aim of this past year's integrated calculation was to help in bridging the vast gap between what can be calculated in a single computation, and what needs to be computed. (Our original goal was to repeat a calculation such as last year's X-ray burst, but for the nova case; we felt, however, that the cellular detonation calculation was better suited to advancing our understanding of the physics of detonations and for validating the 1-D model for the detonation front speed.)

Specifically, several of our target astrophysics calculations (X-ray bursts on neutron stars and Type Ia supernovae in white dwarfs) involve the propagation of a nuclear detonation front. Since these calculations (whose macroscopic scales range from a few kilometers to 10^4 km) cannot hope to resolve the spatial scale of a detonation front (whose detailed spatial structure falls well below a centimeter), one must appeal to a model. The aims of this year's integrated calculation were to carry out direct numerical simulations, in two and three dimensions, of a realistic nuclear detonation front, so that

1. we could demonstrate the functionality of *Flash-1*'s architecture to solve large problems which stress the full resources of one of the ASCI platforms;
2. we could provide challenging 2 and 3-D data sets for state-of-the-art visualization;
3. we could provide a benchmark direct numerical simulation for building a model of nuclear detonation fronts.

The results of these calculations are described in the following section on astrophysics.

3.2 Details of the Calculation

The simulation used 1000 processors of ASCI Blue Pacific in dedicated mode and was completed over a three day weekend. The domain size of the computational grid was $12.8 \times 12.8 \times 256$ cm. The grid size, if fully refined, would be $256 \times 256 \times 5120$ (335 million grid points), which is equivalent to a 700^3 grid. The actual number of grid points at the end of the calculation was 46 million, representing a savings of more than a factor of 7 from using AMR. During the first half of the simulation, the savings was a factor of 40-50.

The amount of data written during the calculation was approximately 1 Tbyte for checkpoint files, and 0.2 Tbytes for plot files. Roughly half of the simulation time was spent doing computation and half doing I/O. We made use of our new parallel HDF 5 I/O routines, which provided a factor of 10 speed up over our previous version. Without parallel I/O, this calculation would have taken three weeks to complete.

At the end of the run, the size of each plot file was 1.2 Gbytes. These files contained a representative sample of the variables used in the calculation (pressure, silicon abundance, and three components of velocity). Writing each file to disk required approximately 5 minutes – a throughput of 4 Mbytes/s. Checkpoint files were written differently. Each processor dumped its portion of the data to its local disk, so that each checkpoint actually consisted of 1000 separate files. This was necessary to avoid the 2 Gbyte file size limitation on Blue Pacific. Checkpointing was performed 100 times during the simulation. Five complete checkpoints were kept on disk at any one time. New checkpoints were then written over the previous ones.

Only the 0.2 Tbytes of plot files were transfered back to Argonne National Laboratory for visualization. This was accomplished with GridFTP (see §5.4), using seven parallel pipes to seven different disks. The transfer speed across each pipe varied from 400 Kbytes/s to 1Mbyte/s. All of the data was retrieved in slightly less than a day.

Details of the astrophysics emerging from this calculation will be discussed in §4.5.

3.3 Emerging issues

Our integrated calculations have raised a number of new issues regarding computing on the ASCI platforms:

1. Non-uniform computational infrastructure. Our strong bias is to construct codes that are as portable as is practicable. Certain features of the existing ASCI computing platforms make this goal difficult to attain, including significant differences in the hardware and the operating system software between the various ASCI platforms. The most important differences relate to

- differences between platforms in how "within box" and "outside the box" calculations are carried out, leading to significant differences in how much (and what kind of) tuning is required to obtain satisfactory "out-of-box" computing performance;
- the need to use threads on some platforms to make efficient use of the full machine, especially for memory-bound problems;
- limits on file size deriving from 32-bit (rather than 64-bit) operating system implementations;

2. Variable site-to-Center communications performance. Our transfer rates from the ASCI platforms to our local machines have been at times disappointing. This is an area in which our Center has received enormous support from scientists at the National Labs; indeed, we have seen substantial improvements within the past 2 months in this regard. An important issue with regard to communications is the progress in establishing Globus as a Lab-sanctioned means of managing computing resources. Thus, ANL and ISI Globus project participants worked with the ASCI DISCOM DRM group throughout the year on technical issues relating to the creation of a Globus-based tri-lab Distributed Resource Management system. This work culminated in successful completion of a number of technical milestones and DOE approval of the DRM security plan, which included the Globus components.

3. Difficulties in visualization. The extremely large data sets we are now producing are clearly stressing the various visualization efforts. In order to visualize our detonation computation, we collaborated with both our FLASH colleagues at Argonne and with the visualization group at LLNL. The latter group used Ensight to create stills and movies of the data; processing each image required 7 Gbytes of memory and 35 minutes of CPU time. Our collaborators at the Futures Laboratory at Argonne National Laboratory are using a somewhat different approach, but are experiencing similar challenges. Even looking at two-dimensional slices on our local machines has proved difficult. We should also point out that this detonation calculation is relatively small when compared to what we hope to simulate in the near future.

4 Astrophysics

Participants: E. Brown, A. Calder, J. Dursi[1], B. Fryxell, R. Krasnoplosky, D. Lamb, C. Litwin, A. Mignone[1], J. Niemeyer, K. Olson, F. Peng[1], P. Ricker, F. Timmes, R. Rosner, J. Truran (Group Leader), N. Vlahakis, Y.-N. Young, M. Zingale

[1] Graduate student

4.1 Mission and goals

The astrophysics group has the responsibility to develop the astrophysically-relevant physics modules for the *Flash* code; to carry out the large-scale astrophysics simulations which are the heart of the FLASH Center; and to carry out the analysis and interpretation of the computational results in light of astrophysical observations.

4.2 Overview of work

The third year of astrophysics research has witnessed significant progress on several fronts. As in the first two years, the early focus was on the X-ray burst problem as the test-bench for development of the various physics modules required for the *Flash* code. The thermonuclear reaction networks, stellar equations of state, and thermal transport coefficient modules that now reside in the *Flash* code have been thoroughly tested and benchmarked, as described and documented in the papers by Timmes [38], Timmes & Arnett [40], Timmes & Swesty [42], and Timmes [39]. Other significant improvements are the inclusion of self-gravity and implicit diffusion. The Poisson solvers added to the *Flash* code have been tested using the Jeans instability problem (for periodic boundaries) and the spherical collapse problem (for isolated boundaries).

The X-ray burst problem also continued to provide a basis for testing the AMR refinement procedures that have been incorporated into the *Flash* code using PARAMESH.

The modifications and improvements to the *Flash* code described in §2 above have allowed us to begin preliminary calculations on all three of our target astrophysics problems, using all available ASCI platforms.

4.3 X-ray burst simulations

We have carried out a wide variety of X-ray burst calculations in order both to understand better the proper use of *Flash-1*, especially its adaptive mesh refinement strategy, and to explore the basic physics underlying nuclear burning on the surface of a neutron star; these basic studies are all preliminary to the eventual full-scale simulation of a neutron star X-ray burst we intend to carry out.

4.3.1 2-D simulation of an X-ray burst

We have carried out two-dimensional simulations of an X-ray burst in cylindrical coordinates. These calculations were performed by Zingale *et al.* [55] on the NIRVANA cluster at the Los Alamos National Laboratory. The initial conditions were chosen to insure the onset of detonation. The evolution was followed, on a scale roughly comparable to neutron star dimensions (1.5 km $\times$ 2 km), on a cylindrical grid (1536×2048 effective grid points) for a total time exceeding 150

μs. This represents an approximately 1 meter resolution. We obtain a velocity compatible with a Chapman-Jouguet detonation. The computation is followed through the point at which the detonation has left the grid. The initial model is a hydrostatic neutron star atmosphere, with a density of 2×10^8 g cm^{-3} at the base, falling off to $\sim 10^6$ g cm^{-3} over 100 meters. Above this, the density falls off quickly to 10^{-5} g cm^{-3}. The atmosphere and material above it are pure helium. We start the burst by increasing the temperature in the lower left corner of the domain to 2.5×10^8 K, at which temperature thermonuclear burning of helium commences, creating a detonation. In the vertical direction, the detonation wave propagates down the density gradient, breaking through the surface of the neutron star atmosphere. Above the atmosphere, the shock detaches from the burning front and races ahead. The detonation front moves along the surface of the star at about 10^9 cm s^{-1}.

The density structure of the envelope at 6 times over the course of the outburst is displayed in Figure 2. In all of these images, a green line marks a helium abundance of 0.9; below this line, burning has begun to deplete the helium. The dark blue line marks 10 g cm^{-3}, giving an indication of how much the neutron star atmosphere has been distorted by the explosion.

Significant results arising from this research included: (i) the detonation moves at the Chapman-Jouguet velocity, 1.3 x 10^9 cm s^{-1}, implying a 3 ms propagation time from pole to pole; (ii) the atmosphere oscillates with a period $\sim$ 50 μs; (iii) the photosphere flows rapidly off the top of the grid at 68 μs, with velocities suggesting a peak height of $\sim$ 10 km; and (iv) a series of surface waves propagate behind the detonation front with a velocity $\sim$ 1.3 x 10^9 cm s^{-1}, consistent with finite amplitude shallow water wave theory.

4.3.2 Ongoing studies

In work related to X-ray bursts, the spreading of accreted fuel (hydrogen and helium) away from the polar cap of a strongly magnetized ($B \geq 10^{12}$ G) accreting neutron star is being studied. The accreted hydrogen and helium ignite where the gas pressure is somewhat less than the magnetic pressure, which motivates the question of how the fuel is actually distributed over the surface when ignition occurs (for an overview of the problem, see [2]). An analytical investigation of the stability of an accreted magnetized mound of material to short-wavelength ballooning modes has been completed [23]. Line-tying in the neutron star's crust and the stratification of the neutron star's atmosphere and ocean stabilize the mound of accreted material until the gas overpressure in the polar cap is sufficiently large. For a realistic model atmosphere we demonstrate that the instability occurs when the overpressure exceeds the magnetic pressure by a factor (several) $\times\, a/h \gg 1$, where a is the lateral length scale and h is the vertical length scale. This instability is expected to produce an enhanced transport of matter across the magnetic field. With the development of a MHD module for *Flash-2*, it will be possible to numerically simulate this spreading.

An investigation of the role of the rp-process (defined by a sequence of (r)apid (p)roton captures onto seed nuclei provided by helium burning; see [4, 50, 49, 32]) in a type I X-ray burst will also be done. For typical conditions, this nuclear processing can produce nuclei with $A > 56$ (as is the case for stable burning; see [34]), and might have implications for energy generation during the late phases of the burst event [31, 21].

An investigation of the role of the rp-process (defined by a sequence of (r)apid (p)roton captures onto seed nuclei provided by helium burning [4, 50, 49, 32]) in a Type I X-ray burst will also be done. For typical conditions, this nuclear processing can produce nuclei with $A > 56$ (as is the case for stable burning; see [34]), and might have implications for energy generation during the late phases of the burst event [31, 21].

4.4 Nova explosions

This year's FLASH Center activities included a concerted effort to understand the physics underlying hydrodynamic thermonuclear runaways on white dwarfs, leading to nova explosions. The most critical question in this regard involves the identification of the mechanism by which carbon, oxygen, and neon enriched matter is dredged up from the underlying white dwarf into the active burning regions of the envelope [46]. One dimensional numerical simulations have confirmed that the detailed features of a nova explosion – e.g the light curve, the energetics, and the composition of the ejected shell – are strongly dependent upon both the time history and the magnitude of such envelope enrichment. The dredge up of carbon, oxygen, and neon to levels ~ 30 % by mass of the envelope [24] allows more explosive hydrogen burning and concomitant energy input on a dynamical timescale. We have begun to address this problem on several fronts.

4.4.1 Exploring the mixing process

A core issue for understanding nova is the extensive observed mixing of stellar material (such as carbon and oxygen) into the burned envelope ejecta; since this material cannot be the result of nuclear burning of the accreted hydrogen/helium envelope, some process of "dredge up" of stellar matter must operate. One of the several possible mechanisms of dredge up [25] that has previously been proposed is shear-induced mixing [19]. The results of this early work unfortunately were inconclusive, and subsequent ideas for mixing by other mechanisms (such as convective overshoot or turbulent erosion) were similarly unsuccessful. We have reexamined this problem with the use of the *Flash* code, based on ideas derived from oceanographic research.

R. Rosner, together with postdoctoral fellow Y.-N. Young and student A. Alexakis, have reconsidered the problem of shear mixing at (density) interfaces in stratified media. In oceanographic work, it has been long recognized that

Kelvin-Helmholtz instabilities cannot account for the observed mixing at ocean or lake surfaces; the focus there has been on instabilities of surface gravity waves driven by an overlying wind. We have now successfully reproduced this work using the *Flash* code, verifying the linear instability, and extending the work into the previously unexplored highly nonlinear regime. In this regime, the unstable surface waves are shown to break, leading to a mixing layer substantially thicker than previously obtained from Kelvin-Helmholtz studies. Our ongoing work is now to incorporate these new results into a model for interface mixing that can be inserted into our full nova calculations.

4.4.2 Nova simulations with the *Flash* code

Two-dimensional simulations are currently being run with the *Flash* code, using a 1-D initial model that has been used for two different sets of multidimensional simulations [11, 16]. These two earlier simulations have given differing answers about dredge up from the white dwarf into the accreted layer. As a first step in our nova studies, we would like to be able to identify and understand the source of this discrepancy. Our simulations, started this summer by J. Dursi, will not only shed light on the difference in results from these two groups, but also serve as a first step towards our future two- and three-dimensional simulations, using different initial models which can help us to understand novae and their observed diversity.

4.4.3 Nova simulations with ODT

The One Dimensional Turbulence (ODT) model, developed by Alan Kerstein at Sandia National Labs in California, has been successfully used to model mixing in many physical systems. Pre-runaway mixing in a nova can serve to dredge up material from the white dwarf, which will crucially affect the runaway evolution; since examining large numbers of different mixing scenarios with *Flash* code simulations is prohibitively expensive, we have chosen to use ODT as a method for exploring the dependencies on our initial model assumptions.

ODT, as originally formulated, does not include gravity as a dynamic effect, nor multiple species nor energy source terms. This summer, these effects were added to a version of a code which implements ODT, and initial experiments were undertaken with modeling dredge-up from the white dwarf's surface. These calculations complement the direct numerical simulations of gravity wave breaking discussed just above.

4.5 Supernova Ia explosions

Progress has also been made in our efforts to understand the physics of Type Ia supernova explosions. We have chosen to focus our attention on the manner in which the burning regimes of the nuclear flame can provide a clear and consistent picture of the stages of the explosion. It is an understanding of the evolution from the early flamelet regime to the distributed burning regime and, ultimately, to a possible deflagration-detonation transition - the microphysics of flames - that is essential to the formulation of realistic sub-grid models for the behavior on small scales. We have studied several aspects of this problem.

4.5.1 Cellular structure of carbon detonations in two dimensions

While there have been a number of experiments and numerical studies for detonations occurring in terrestrial materials, the role of the cellular structure of detonations in astrophysical applications to Type Ia supernovae has not yet been fully explored. Issues of interest include: (i) the degree to which the resolution required to reveal the cellular structure can act to define the minimum resolution required for multidimensional simulations of detonations in Type Ia supernova models and (ii) the implications of such structures for the spectra and nucleosynthesis contributions of supernovae. In the context of our ASCI studies and goals, we were concerned with whether the resulting cellular structure might give rise to levels of chemical inhomogeneity in the detonated matter that could provide constraints upon the character of the burning history.

Timmes et al. [44] have performed two-dimensional simulations of carbon detonations for conditions that are compatible with the results of one-dimensional models of Type Ia supernova events, with an initial (upstream) density of 10^7 g cm^{-3}. While such features as the curvature of the weak incident shocks, the strength of the triple points and transverse waves, and the sizes of the under-reacted and over-reacted regions at this density were found to depend strongly on the spatial resolution of the calculation, this was not true of the cell sizes. Rather, we found that the cell sizes of a two-dimensional detonation propagating through pure carbon at this density are robust with respect to the spatial resolution of the simulations. The cellular instabilities result in pockets of incompletely burned material, and this produces a somewhat different composition distribution in the detonated material than that resulting from one-dimensional calculations. Since the cell sizes we obtained are significantly smaller than the pressure scale height in a white dwarf at this density and composition, it is unlikely that this particular cellular structure would lead to observable levels of inhomogeneity in supernova spectra or nucleosynthesis products; however, at densities of 10^6 g cm^{-3}, the cell size can become comparable to the star size; and in that case, observable effects may well occur. Calculation of this effect clearly lies on our (astrophysics) roadmap.

4.5.2 Cellular structure of carbon detonations in three dimensions

We have also carried out a three-dimensional simulation of a carbon detonation [45] , for the same initial conditions as described above for the two-dimensional case. This was a large integrated calculation, carried out on 1000 processors on ASCI Blue Mountain, at LLNL. (Details of the simulation are described in §3 above.) An obvious question here is whether there might be significant differences between the 2D and 3D cellular structures of carbon detonations. As for the 2D case, we found strong dependences upon the spatial resolution (and dimensionality) of the calculation. The strong symmetries that are present in the two-dimensional simulations are weakened or entirely absent in three dimensions. The distribution of the silicon ashes produced behind a detonation front formed by a supernova explosion is displayed in Figure 4. The three-dimensional structure of the front results in pockets of unburned material and a slight reduction in the propagation velocity of the detonation. As with the 2-D simulation, the scales of what features persist are small with respect to a pressure scale height, it would appear unlikely that variations in composition between under-reacted and over-reacted regions will impact either the nucleosynthesis yields or spectral features of supernova explosions.

4.5.3 Quenching of thermonuclear flames

Thermonuclear burning in a Type Ia supernova begins as a flame, deep in the interior of a white dwarf. Scrutiny of supernova spectra suggests that, at some point, the burning may undergo a transition from a deflagration to a detonation. Some mechanisms for this transition require a preconditioned region in the star. As the flame propagates down the temperature gradient, the speed increases, and the transition to a detonation may occur[18, 27]. For this to happen, the region must be free of any temperature fluctuations. Any burning that was occurring in that region must be quenched.

The existing *Flash* code is giving flame speeds which are consistent with those computed by Timmes & Woosley [43]. This is an important validation of our code, and an essential step on our way to a more realistic treatment of the supernova problem.

We have begun direct numerical simulations of flame-vortex interactions, in order to understand quenching properties of thermonuclear flames. A key question is whether a thermonuclear flame can be quenched. If not, the DDT mechanisms that demand a finely tuned "preconditioned" region are unlikely to work. In our simulations, we pass a steady-state laminar flame through a vortex pair. The vortex pair represents the most severe strain the flame front will encounter inside the white dwarf. We vary the speed and size of the vortex pair in order to explore the characteristics of the quenching process as a function of stellar properties. This research is currently in progress with the *Flash* code.

4.5.4 Combustion and turbulence

Two-dimensional simulations have been run of a turbulent region of white-dwarf material being spherically 'imploded', eventually self-igniting. Since the evolution of the turbulence under the homogeneous compression can be understood by other means (using Rapid Distortion Theory), the evolution with combustion gives us an understanding of how the turbulence and combustion interact. In another set of experiments, a self-sustaining thermonuclear 'flame' is run through a turbulent region. This both gives us insight as to how laminar flames and turbulence interact, and also allows us to begin constructing a flamelet 'subgrid model' for use in the burning in a Type Ia supernova.

4.5.5 Subgrid models

For the *Flash* code to be able to use a subgrid model for the evolution of a flamelet through a supernova type Ia progenitor, it must know accurately where the flame is; however, we will never be able to have enough resolution to evolve the flame itself. Thus, we must use some sort of interface-tracking method to follow the flame's progress. This is greatly complicated by the parallel, adaptive, multidimensional nature of the code, and the fact that we expect the flame front to go through complex changes in topology during its evolution.

A variant of the Level Set Method algorithm which overcomes these difficulties has been developed for the *Flash* code, and is being implemented.

4.6 Generally-applicable MHD effects

We have been working on physics studies of the circumstances under which accretion onto magnetized compact objects (neutron star or white dwarf) occurs. One central question is how the accreted material is "placed" on the stellar surface: does the accretion occur primarily at the poles, or is the material more uniformly spread over the surface? Work by C. Litwin, R. Rosner, and D.Q. Lamb [22] has shown that the answer seems to depend on the geometry of the accreting stream: If the stream is well-collimated, then it is possible that accretion occurs only over a small portion of the stellar surface, which may not even be at the poles. In more recent work, C. Litwin, E. Brown, and R. Rosner have examined the stability of accretion columns on neutron stars, asking under what circumstances magnetic fields may prevent the spreading of material over the stellar surface [23] and have obtained estimates for the onset of instability (due to unstable ballooning modes).

4.7 Workshops

Significant interactions with the nova, X-ray burst, and supernova communities over the past year have also provided important input to ASCI research. A Workshop on 'Astrophysical Thermonuclear Explosions,' was organized by E. Brown, J. Niemeyer, R. Rosner, and J. Truran in June 2000. For this workshop, we were fortunate to have been able to bring together, in a rather unique setting, leading researchers from each of these diverse fields. Nova research was discussed in contributions by M. Livio, J. Truran, S. Starrfield, A. Glasner, A. Kercek,

R. Williams, and R. Gehrz; the X-ray burst phenomena by L. Bildsten, M. Wiescher, W.R. Hix, R. Sunyaev, M. Zingale, T. Strohmayer, and A. Cumming; and supernovae by S. Woosley, B. Leibundgut, P. Höflich, M. Reinecke, and J. Niemeyer. Proceedings of this workshop are expected to be published in early 2001.

4.8 Interactions

The Astrophysics group has collaborated with scientists both at the Labs and at other universities; collaborators include:

1. D. Arnett (supernovae, validation; University of Arizona/Tucson)
2. A. Bayliss (novae and X-ray bursts; Northwestern University)
3. A. Burrows (supernovae; University of Arizona/Tucson)
4. R. Eastman (radiative transfer, supernovae; LLNL)
5. A. Glasner (novae; Hebrew University of Jerusalem)
6. W. Hillebrandt (novae and supernovae; MPI Garching bei München)
7. R. Hoffman (reaction networks; LLNL)
8. D. Lin (novae and X-ray bursts; Northwestern University)
9. E. Marietta (supernovae; University of Arizona/Tucson)
10. E. Müller (relativistic astro; MPI Garching bei München)
11. T. Plewa (supernovae; MPI Garching bei München)
12. D. Swesty (radiative transfer; SUNY at Stony Brook)
13. R. Taam (novae and X-ray bursts; Northwestern University)
14. S. Woosley (supernovae and X-ray bursts; University of California at Santa Cruz)

5 Computer Science

Participants: A. Chan, T. Clark, P. Fischer, J. Flaherty, I. Foster, L. Freitag, E. Gomez[1], W. Gropp, R. Hudson, J. Hensley, R. Loy, E. Lusk (Group Leader), S. Meder, M. Papka[1], J.-F. Remacle, P. Ricker, R. Scott, M.S. Shephard, M. Singer, R. Stevens, R. Thakur, H. Tufo, T. Udeshi

[1] Graduate student

5.1 Mission and goals

The Computer Science research component of the FLASH Center is carried out in multiple interrelated areas, including Numerical Algorithms and Methods, Software architecture and design, Scientific Visualization, Distributed Computing, and Scalable Performance and I/O. These are the fundamental research areas on whose results the *Flash* code development effort is, and will be, based.

We note that because the interactions between Center computer scientists and computer scientists at the DOE National Laboratories are so extensive, we have not called out these interactions in a separate section; instead, we mention these interactions as part of the following discussion of our studies and results.

5.2 Numerical algorithms and methods

In this area we seek to develop scalable numerical methods, solvers, and libraries for scientific simulations. During the last year, we have been conducting a number of experiments on all of the ASCI machines and similar supercomputers. In particular, we have carried out two- and three-dimensional Rayleigh-Taylor simulations (on O2K, ASCI Red, ASCI Blue Mountain), two- and three-dimensional forced convective heat transfer in grooved and grooved-flat channel simulations (on O2K and Intel Paragon), buoyant convection in a rotating hemispherical shell simulations (on O2K, T3E), and hairpin vortex simulations (on O2K, ASCI Red).

5.2.1 Spectral Element calculations

Our code implementation milestones include a spectral element code for multi-million gridpoint simulations of incompressible flows in general two- and three-dimensional domains which now runs on all three ASCI platforms, as well as on (for example) the T3E, IBM SP-2, O2K, and NOW. We have achieved 319 Gigaflops on 2048 nodes of ASCI Red for one of the hairpin vortex simulations. This code uses MPI/NX for internode and OpenMP for intranode parallelism, thus exploring mixed-mode computations. Further advances along this line include extension to solving the anelastic equations (e.g., equations allowing for compressibility effects at very low Mach number), in collaboration with the Astrophysics and Code Groups (cf. §3-4).

One area of particular interest is development of scalable solvers for elliptic problems. Towards this goal, we have developed a parallel direct solver for solution of the coarse-grid problem that readily scales to thousands of processors [47].

We have interacted with all three of the ASCI labs. In particular, we have worked with A. Cleary at LLNL to put our parallel direct solver into HYPRE, worked with R. Tuminaro at SNL to put our parallel direct solver into ML, and (as part of a collaboration with the Code and Validation/Basic Science Groups) worked with B. Benjamin at LANL this fall to simulate gas curtain experiment with the *Flash* code.

5.2.2 Discontinuous Galerkin (DG) techniques on unstructured and structured meshes

Our goal is to determine, first, if the DG method offers any advantages over other high resolution methods on structured meshes, and second, if unstructured meshes can compete with structured meshes for the physics of interest (in this case Rayleigh-Taylor instability simulations).

The advantage that DG offers is a more compact stencil, which might result in more efficient use of refinement and better message passing efficiency; possible disadvantages are that the method may require a smaller time step to maintain stability compared to PPM, may require more memory, and may require more work/cell for the third order method due to a large number of required flux computations. We have now carried out a number of simulations (of the canonical RT problem, using the same initial and boundary conditions used by our *Flash-1.x* simulations) using DG on both structured and unstructured meshes.

5.2.3 Software architecture and design

Our goal is to investigate the fundamental concepts in creating a component-based model for numerical and application codes (with the *Flash* code being a primary motivator).

With this goal in mind, we are developing a testbed environment to enable analysis of the tradeoffs in performance and flexibility associated with component-based architectures. We developed a generic mesh and field data interface for the discontinuous Galerkin solver implemented last year in SUMAA3d. This interface separates the application code from the specifics of the SUMAA3d data structure and will ease migration of the solver to other mesh management packages. Ongoing work includes tuning the interface to achieve better performance, and working with computer scientists in the CCA forum to develop an interface standard for mesh and data objects. We have also, in collaboration with the Architecture Team of the Code Group, started an investigation of concrete implementations of mesh components for *Flash*; this work is relevant not only to understanding how DG (which is currently implemented using RPI's Trellis) can be used within *Flash*, but is also relevant to the issue of "importing" the spectral element module into *Flash-2*.

5.3 Scientific visualization

5.3.1 Visualization infrastructure

Building from the efforts of the previous two years, we have improved the infrastructure for the creation of high-resolution movies of the FLASH Center's two-dimensional simulations. This procedure now is completely automated and the infrastructure supports the creation of movies for arbitrary tiled display configurations, meaning the same infrastructure has produced movies for Argonne's ActiveMural and microMural displays as well as University of Minnesota's Powerwall. We have also made efforts to improve playback performance of movies on tiled displays and have added more user controls to the playback tool.

5.3.2 Visualizing multiresolution data sets

Last year's three-dimensional simulations required a resampling of the data from its multiresolution form to a uniform mesh for visualization. This year we are able to visualize the FLASH Center's three-dimensional simulations in their native format, directly from either HDF-4 or HDF-5. The key component is the generation of a set of *Flash* HDF C++ classes, written in collaboration with the *Flash* code team that allow for reading directly from *Flash* output files into the various visualization tools. The *Flash* HDF classes provide a layer of abstraction between simulation output and visualization input. Based on the *Flash* HDF infrastructure and the Visualization Toolkit (vtk), we have been able to construct visualization tools specific for the *Flash* datasets. These tools are capable of displaying cutting planes, wire-frame representations of simulation grids, and isosurfaces, from the native multiresolution *Flash* dataset. The visualization tools exploit the multiresolution nature of the data, allowing for faster renderings of the dataset by using the data at lower levels of refinement. Additionally building from last year's efforts in the two-dimensional datasets, the notion of trails has been extended to the three-dimensional datasets. A trail is a preprocessing step that passes over the data without the use of graphics hardware to construct a path (or trail) through the data featuring points of interest. These trails then provide the visualization component with enough information that the dataset can be visualized using a fraction of the data, yet providing the required or interesting parts for presentation.

5.3.3 Parallel Volume Rendering

Volume rendering is a natural technique for visualizing the FLASH Center's three-dimensional datasets, and this was demonstrated at the Site Visit last year within the CAVE environment. The problem was that the dataset needed to be resampled at the highest level of refinement on to a regular grid. This is even true of the desktop utilities that the FLASH Center itself uses for debugging and visualizing the output. This year we have constructed a scalable distributed volume-rendering tool. This tool works with the FLASH Center's datasets in their native format, both HDF-4 and HDF-5, as well as regular grids. The parallel volume renderer was designed to use MPI and runs on Argonne's Chiba City cluster, but should be portable to any platform. The tool has been modularized so that data loading, rendering, and compositing can all be scaled as needed. The code has been tested using 64 of Chiba City's 256 dual processor nodes, running with two processes per node.

5.3.4 Vector visualization using line integral convolution

An additional Argonne effort has been in the area of vector visualization of the two-dimensional simulation datasets. We use the Line Integral Convolution (LIC) visualization algorithm to display two-dimensional, multi-resolution vector fields. (LIC allows the viewer to see local and global properties of the vector field as a texture image.) We've implemented a version of LIC – the "FastLIC" algorithm – and adapted it to work with multi-resolution grids. FastLIC uses three grids: a mono-resolution noise image as input, the data grid as input and a mono-resolution texture image as output. It treats these three grids' resolutions orthogonally. This allows us to create output images of any resolution, regardless of the data grid resolution-including, in our case, multiple resolutions. The only place the data grid resolution becomes important is during bilinear interpolation of the vector field to determine vector values at exact output image pixel locations. We also color-map the LIC texture according to a scalar field derived from the vector field – usually the vector magnitude.

5.4 Distributed computing

We have been working on a number of barriers to remote access to experimental data sets. The FLASH project provides a good example on which to work on, since the calculations are often being done at the ASCI labs, while we wish to examine the results at Chicago and visualize them at Argonne.

5.4.1 Data Grid

Access to distributed data is often as important as access to distributed computation resources. Distributed scientific and engineering application typically require access to large amounts of data (terabytes or petabytes). Collaboration surrounding such applications requires widely distributed access to data. The distributed scientific computing community has envisioned a number of strategies for supporting these application needs, which have collectively come to be known as the Data Grid.

We are currently engaged in defining and developing the following core capabilities which we believe will be necessary in order to build a persistent Data Grid environment.

- GridFTP: A high-performance, secure, robust data transfer mechanism
    * Automatic negotiation of TCP buffer/window sizes
    * Parallel data transfer

  * Third party control of data transfer
  * Partial file transfer
  * Security
  * Support for reliable data transfer

- A set of tools for creating and manipulating replicas of large data sets
- A mechanism for maintaining a catalog of dataset replicas

5.4.2 Data Grid and *Flash*

The FLASH project is an excellent example of the data-intensive applications mentioned above. We are applying Data Grid technologies to the FLASH project in the following ways:

- Use GridFTP tools to transfer data sets (200 GB+) from ASCI Centers (LLNL) to a Data Grid cluster at Argonne National Laboratory
- Provide FLASH scientists with GridFTP clients to access the data at Argonne
- Develop a GridFTP file driver (with partial file transfer) for HDF-5
- Encourage the use of GridFTP/HDF 5 for data visualization applications
- Explore the usefulness of replica sites at other locations and build a replica catalog for replicated *Flash* datasets

Recent work has included the transfer of an initial 200 GB data set from Lawrence Livermore to Argonne, development of GridFTP libraries and tools, and performance testing for the GridFTP protocol.

5.4.3 Beta Grid

Emerging "Grid" applications such a Science Portals, Data Grids, and large-scale parameter studies require on-demand access to computing and storage capabilities. Unfortunately, the supercomputing resources that we have traditionally relied upon for cycles and storage are not designed to support such on-demand access. We believe that the solution to this problem is to populate our networks with low-cost compute-storage clusters loaded with standard software for remote access scheduling, management, accounting, and the like. These Beta Grid Nodes (BGNs) will serve as power plants for the Grid, supporting applications that need on-demand access to a few minutes or hours of compute power or a terabyte of disk space. The BGN software suite will include support for high-performance data transfer via striped FTP, reservation of disk space and CPU and policy-driven access control.

Applications that can exploit BGN capabilities include:

- BGN-enabled Access Grid Data Visualizer
- BGN-enables Portals (Web-based user interface to simulation codes)
- BGN-enabled parameter studies
- Network monitor

5.4.4 Beta Grid and *Flash*

We have started what we envision as a streamlined, community-based effort aimed at creating and deploying a BGN infrastructure during 2001. We already have a prototype BGN operational and will be developing further essential software in the next few months. We intend to apply Data Grid technologies to *Flash* in the following ways.

- Encourage the development of visualization code that can do its computation on a beta grid platform by decoupling data access and computation from the display.
- Demonstrate that FLASH visualizations can be requested from desktop computers, then created on beta grid nodes for display on the desktop.

5.5 Scalable performance and I/O

We have worked with IBM and Lawrence Livermore National laboratory to implement a tool for the graphical display of parallel program behavior, especially in the case of log files for large runs produced by IBM's event-logging tools [52]. The SLOG (Scalable Log File) tool was delivered to IBM, who deployed it on the ASCI machines at Livermore. The Jumpshot tool for displaying such log files was further enhanced and can also function with log files created via the MPI profiling interface and the MPE library, which is portable. Thus logfile collection is more efficient on IBM platforms, where our tolls interface directly to the AIX tracing mechanism, but the system is portable to all of the ASCI parallel computing platforms and applicable to all programs that use MPI.

We released an update of the ROMIO portable implementation of the MPI-2 standard for parallel I/O. It is now in use at all of the ASCI labs, and provides important support for the HDF-5 library from NCSA, used by the *Flash* code and by ASCI program codes.

The scaling tests performed by the Code Group were carried out in collaboration with members of the Computer Science Group. These scaling tests were conducted on the ASCI machines and others using well-understood instantiations of the *Flash* code. Results of these tests were presented at the interim site review in the summer and more results, on a benchmark that is more difficult to scale effectively, will be presented at the October site review. These results have provided insights into comparing the ASCI machines with one another and to other large-scale machines, as well as helping us understand scaling issues in the *Flash* code itself.

We have conducted experiments with a prototype of topology-aware collective MPI operations, particularly at Livermore, where there can be a three-level hierarchy of communication performance in a single MPI job. Our experiments have convinced us that we should incorporate these types of optimizations into our next-generation MPI implementation [15].

We have designed an even more efficient and scalable file structure SLOG-2 [5] for scalable log files (those that can be effectively displayed by a Jumpshot-like tool regardless of size) and plan to work with IBM and Livermore in the coming year to implement and test it and utilize it in the performance analysis of *Flash* and other codes.

5.6 Software support for unstructured computation

We address compiler and runtime support for unstructured and irregular applications, in particular in situations where the computational load and possibly the computation to be performed is dependent on the input data and the evolution of the problem during computation. Runtime support is required since the information required to distribute the load and computational work is known only at runtime. We have found that static analysis is also required to insert required calls to the runtime system in the program source code.

In the past year we have further developed the Streams, Overlapping and Shortcutting (SOS) library. SOS supports an overlapping technique, which includes out of order dynamic message scheduling to tolerate load imbalances. It also supports Pstreams, which allow task parallelism in which implicit process groups are determined by program logic at runtime, and shortcutting, which in some situations allows the exploitation of computational asymmetry to speed the computation [12].

Testing is being done using the SOS library in real scientific applications, and the robustness of the library has been improved. We have also seen the need for, and incorporated, additions to the API in support of more flexible communication of parts of arrays. Details may be found at the web site http://people.cs.uchicago.edu/~ernesto/sos.html

We have developed Pstream theory to support automatic program analysis to determine where implicit process groups split into subgroups and merge into larger groups; this theory also determines what conditions are required to insure determinism and freedom from deadlock in such a program run. We have also identified the analysis required for insertion of calls to support overlapping in a program.

6 Validation and Basic Science

Participants: A. Alexakis[1], G. Bal, A. Calder, F. Cattaneo, P. Constantin, J. Curtis[1], D. Dawson[2], A. Draganescu[1], T. Dupont (Group Leader), J. Dursi[1], B. Fryxell, D. Grier, R. Grigoriev, C. Josserand, L. Kadanoff, A. Kiselev, T. Linde, C. Litwin, Y. Liu[1], A. Malagoli, M. Medved[1], Q. Nie, A. Oberman[1], R. Rosner, O. Ruchayskiy[1], L. Ryzhik, R. Scott, H. Tufo, N. Vladimirova, Y.-N. Young

6.1 Mission and goals

Our Validation & Basic Science Group has focused on a variety of fundamental physics problems, including mixing, combustion, turbulence, the motion of interfaces, and multi-diffusion. The aim is two-fold: first, we seek to understand basic physical processes relevant to the FLASH Center problems in order to construct reliable computational models (for example, of unresolved flames); second, our computational tools must be validated by comparisons with laboratory experiments, and in order to carry out such comparisions, we need substantial understanding of the underlying basic physics. It is noteworthy here that a number of the issues we have identified as central to the FLASH Center are also of considerable interest to the larger ASCI program as a whole.

6.2 Rayleigh-Taylor and Richtmyer-Meshkov instabilities

A key problem for our astrophysics applications is that we do not understand how chaotic flows within the star affect the propagation of deflagration fronts. Convective instabilities in the burning region and Rayleigh-Taylor and Kelvin-Helmholtz instabilities along the burning front can all affect the propagation speed by stretching the flame front and by introducing small-scale turbulent mixing and energy transport, which may dominate molecular diffusion processes [17, 18]. However, there is no hope that the deflagration front for a Type Ia supernova calculation can be resolved on a grid which simulates the behavior of the entire star. One reasonable approach is to do a high-resolution simulation of a small section of the burning front, in order to obtain its speed, and then to use the result as a parameter in the full model, combined with a front tracking method.

We therefore started out by aiming at a variety of mixing problems, including convective mixing, mixing in a flame front, and Rayleigh-Taylor and Richtmyer-Meshkov mixing. The latter two problems provide an especially good opportunity to use both historical and newly-generated data. The experimental program has a strong collaborative component with the National labora-

[1]Graduate student
[2]Summer student, now at UIUC

tories, including work with G. Dimonte (LLNL; Rayleigh-Taylor), B. Remington (LLNL; Rayleigh-Taylor, Richtmyer-Meshkov), and B. Benjamin (LANL; Richtmyer-Meshkov). As part of this program, we sent two graduate students to LLNL two summers ago (one working on data analysis, the other working on simulations); and this past summer again sent two students out to Sandia/Livermore (to work on mixing and flame models). The mixing effort is a broad collaboration between Chicago experimentalists, theorists, and computational physicists, including a dozen or so students and postdocs together with S. Wunch and A. Kerstein (Sandia/Livermore) and people in the CNLS at LANL. One particular success is that a simplified mixing model, pioneered by Kerstein, was further developed and tested at Chicago with good agreement between results at Chicago and Livermore. In collaboration with G. Dimonte, we are participating in a consortium of experimentalists, theorists/modelers, and computational physicists to focus on the Rayleigh-Taylor problem; the first consortium meeting took place Oct. 30, 1998; the second took place Oct. 11-12, 1999; and a third is tentatively scheduled to take place in February 2001.

In order to carry out this program, one of our foci has been the (nonlinear) development of the Rayleigh-Taylor instability. There are two specific questions we seek to understand:

Does the nonlinear evolution of the Rayleigh-Taylor instability lead to significant flame front stretching?

Does Rayleigh-Taylor mixing lead to a significantly enhanced effective heat and mass diffusivity?

Our "stable" of distinct types of hydrodynamic codes we can use to answer these questions include a pseudospectral code, a spectral element code (both of which are useful for solving weakly compressible problems) and the fully-compressible *Flash* code. Thus, what we do is:

1. Carry out direct numerical simulations for well-defined weakly compressible problems that have available experimental data, using the two distinct spectral codes; a JFM paper has been submitted on this work [53], which (among other things) shows that we obtain the same results for the integral scales of the flow to within 1-2%.

2. Carry out direct numerical simulations of both the weakly compressible problem and the more compressible (larger Atwood number) problem using the *Flash* code and compare the results with both experimental data and results obtained from other (compressible) codes.

We are now in the process of carrying out a full grid of compressible calculations for both single-mode and multi-mode perturbations in both 2 and 3-D. We have also (in collaboration with B. Remington and J. Kane of LLNL) carried out RT and RM calculations for a multiple-layer laser target and are comparing the results of our calculations both with experimental data obtained at the Omega laser and with simulation results obtained by other codes.

Finally, we have initiated a collaboration with the Benjamin group at LANL, who operate a Richtmyer-Meshkov instability experiment using a "gas curtain" flowing within a shock tube. We have obtained preliminary results for this problem with *Flash*.

6.3 Speed-up and Quenching of Flames

In previous research we focused on the following question: What characteristics of the ambient fluid flow are responsible for burning rate enhancement? The question needs first to be made precise, because the reaction region may be complicated and, in general, may move with an ill-defined velocity.

In order to resolve this ambiguity, P. Constantin [7] defined a quantity V which formally represents the bulk burning rate (which is itself always perfectly well-defined). One can then derive explicit estimates of V in terms of the magnitude of the advecting velocity and the geometry of streamlines. In situations where traveling waves are known to exist, V coincides with the traveling wave speed and the estimates thus automatically provide bounds for the speed of the traveling waves. For a very general class of advecting velocities, it can be shown that the bulk burning rate may not exceed a linear bound in the amplitude of the advecting velocity. The main result of [7] is the identification of a class of flows that are particularly effective in speeding up the bulk burning rate. The main feature of these "percolating flows" is the presence of tubes of streamlines connecting distant regions of burned and unburned material. For such flows we obtained an optimal linear enhancement bound $V \geq KU$ where U represents the magnitude of the advecting velocity and K is a proportionality factor that depends on the geometry of streamlines but not the speed of the flow. The result holds if the velocity spatial scales are not too small compared to the reaction length scale, and the time scale of change of the advecting velocity is not too small compared to a time scale associated with the laminar velocity and the width of the coherent tubes of streamlines (for example, see [54]).

The result applies to a large class of flows that are not necessarily spatially periodic nor sheared and can have completely arbitrary features outside the tubes of streamlines. The bulk burning rate is still linear in the magnitude of the advecting velocity, no matter what kind of behavior (closed streamlines, areas of still fluid, etc.) the flow has outside the tubes. The proportionality coefficient depends on the geometry of the flow in a rather complex manner. These results have been tested numerically, and agreement with the theoretical prediction is very good.

Other flows, and in particular cellular flows, which have closed streamlines, produce a weaker enhancement [20]. Roughly speaking, in terms of their geometry and burning enhancement properties, such flows can be thought of as the opposite of the percolating flows. One can expect the burning enhancement to be significantly weaker for cellular flows, because of the numerous diffusive interfaces which prevent hot and cold regions from mixing fast. Cellular flows pose a mathematically more challenging problem because of these diffusive interfaces. The estimates for percolating flows form only a fraction of the argument we need in the cellular case. In the large flow regime, we obtain the estimate $V \geq CU^{1/5}$. Numerical experiments confirm the sublinear behavior of $V(U) \sim CU^{\alpha}$ for the cellular flows with $\alpha \approx 0.25$.

More recently we turned to a question related to quenching. We started our investigations with a simple scalar reaction-advection-diffusion equation with ignition-type nonlinearity and discussed the following question: What kinds of velocity profiles are capable of quenching any given flame, provided the velocity's amplitude is adequately large? Even for shear flows, the answer turns out to be surprisingly subtle.

If the velocity profile changes in space so that it is nowhere identically constant (or if it is identically constant only in a region of small measure), then the flow can quench any initial data. But if the velocity profile is identically constant in a sizable region, then the ensuing flow is incapable of quenching large enough flames, no matter how much larger is the amplitude of this velocity. The constancy region must be wider across than a couple of laminar propagating front-widths. The proof uses a linear PDE associated to the nonlinear problem and quenching follows when the PDE is hypoelliptic. The techniques used allow the derivation of new, nearly optimal bounds on the speed of traveling wave solutions. Understanding the issues of quenching for cellular and more general flows is a natural next question. Clearly, just as in the case of speed up, the outcome depends on the nature of the flow. We hope to make contact with Navier-Stokes velocity fields and study certain physical models of coupling between the reacting species and the fluid flow.

6.4 Interface motions

We have a general interest in being able to model sharp fronts efficiently and accurately. The understanding that we develop in studying the motion of interfaces should, and we expect will, play a useful role in the FLASH project. The separation of interface motion from mixing is not completely sharp, but roughly speaking the research in this section deals with (relatively) stable fronts.

We ran a workshop this last summer on the topic of interfaces and had several scientists visit and share their expertise with us on this topic. We used funds from a variety of source to leverage the ASCI support.

A group of us has been working on a potential weak point in simulations of such flows as in Rayleigh-Taylor systems. The problem is twofold:

a. The standard fluid mechanics equations do not have meaningful solutions when the surface tension is zero, and

b. These equations produce singularities which slow down and hamper calculations.

In order to avoid computer and numerical artifacts, we are carrying on this work in close collaboration with the Materials-Lab-funded group of Professor

Nagel. His group does experiments on cylindrically symmetrical problems which go to singularities. Roman Grigoriev has built a code which simulates these systems. The comparison with experiment and among theories is quite good, but we need to make the code more efficient to study the details of interest. The program that we are concentrating on at present is a Finite Element code for simulating axisymmetric free surface multifluid Navier-Stokes flows in stream function variables, which employs the mixed Eulerian-Lagrangian formulation for the dynamics of the fluids and the interface.

Sharp interfaces frequently form in situations which involve more than one diffusion. In the FLASH problem there are many situations that involve diffusion of different quantities and each typically has its own rate; these rates can be vastly different. One approach that is being used to develop our skills in these situations is to model some ASCI-supported table-top experiments. These experiments are being done by Jennifer Curtis, a student of David Grier.

A code has been developed by Andrei Draganescu, a student of Todd Dupont, to model the experiments. This code, which was developed with the help of several researchers at ANL, treats two- and three-dimensional problems; it is based on a Boussinesq formulation as the mathematical model of the physical experiments. The code is being tested and refined, and the experimental results are still quite preliminary, so no direct comparisons have been made.

There has also been theoretical numerical analytic work done on related problems. This involves convergence results for a robust technique for time-discretization error control, and the study of the behavior of Galerkin and mixed methods which use moving meshes.

6.5 MHD

Validation of MHD effects is extremely challenging. The problem is that most laboratory experiments on conducting gases or fluids do not operate in astrophysically-relevant regimes: for example, most hot plasma experiments generally do not even operate in regimes which are fully collisional, so that the applicability of single-fluid theory (and related equations) is highly suspect. This problem is particularly accute for problems in which dissipative effects may be important, since it is usually the case that these effects dominate at small spatial scales (on which the plasma is most likely to be collisionless).

For this reason, C. Litwin and R. Rosner have initiated discussions with the experimentalists at both LANL and SNL involved in the Z-pinch experiments, which have the highest probability of providing constraining experiments under conditions in which single fluid equations are applicable. In addition, we have discussed possible validation comparisons with experiments at the Princeton Plasma Physics Laboratory (with M. Yamada, W. Tang, and N. Fisch), based on both plasma and conducting fluid (liquid metal) experiments. These contacts will become more important after our MHD module is integrated into *Flash-2.*

6.6 Interactions and collaborations with the Labs

We have a regular program of exchange with LLNL, LANL, and Sandia/Livermore in the area of Validation and Basic Science.

Leo Kadanoff has for some time had a working relationship with A. Kerstein, of Sandia National Laboratory. In the last year, we have kept this relation ongoing by meeting here twice and twice in Livermore. One important bridge to his group continues to be Dr. S. Wunsch, who obtained his PhD with Kadanoff at Chicago, and has been working in Kerstein's group ever since.

Another form of interaction is via seminars. The Computations in Science seminar (co-sponsored with the Computations Institute) regularly invites speakers from DP labs, and we also visited extensively at the DP labs.

The particular collaborations are as follows:

LLNL:

- G. Dimonte, A. Cook et al.: LEM experiments, Rayleigh-Taylor instabilities, "Alpha Group"
- B. Remington's group: Rayleigh-Taylor and Richtmyer-Meshkov instability experiments on Nova, Omega, and NIF lasers; calculations of instabilities in supernovae

LANL:

- R. Holmes: Comparison of Rayleigh-Taylor and Richtmyer-Meshkov simulations to laser experiments
- M. Gittings and B. Weaver: RAGE code
- B. Benjamin's group: gas curtain experiments
- J. Kamm and B. Rider: simulations of gas curtain experiments

7 Leveraging

A substantial number of our activities take advantage of other related (non-ASCI) projects carried out by scientists affiliated with our Center. Examples include

- The Argonne Mathematical and Computer Sciences group carries out a large number of non-ASCI supported activities directly related to our Center; outstanding examples include work on MPI, MPI-IO, mathematical libraries, and advanced visualization.

- Activities in the Chicago Material Research Sciences Center play an important role in our Validation and Basic Science program. Examples include the experimental and theoretical work on interfaces; studies of models for turbulence; experiments and theoretical work on double diffusing systems; and work on level set stretching.

- Computational physics work carried out as part of NASA-supported activities at Chicago, including work on pseudospectral codes and incompressible MHD, has played important roles in assisting studies carried out as part of our Center activities.

- The core adaptive mesh refinement package used by the present versions of *Flash*, PARAMESH, is a software project whose origins are at the Goddard Space Flight Center, where its development was initially supported by NASA; NASA in fact is continuing this support at an enhanced level.

- The University of Chicago is a partner in the National Partnership for Advanced Computational Infrastructure (NPACI), and this activity (which supports T. Clark) has provided additional expertise in parallel computing that has been useful to FLASH.

8 Personnel

The third year has seen a substantial increase in our complement of scientists involved in the FLASH Center; a full listing of the scientists and support staff supported fully or in part by the Center is provided in the table shown in the Appendix below.

8.1 New hires

In FY00, we have hired two young experts in MHD, R. Krasnopolsky and N. Vlahakis; the latter is also a (prestigious) McCormack Fellow in the Enrico Fermin Institute. We are also planning to hire one additional young computational scientist, T. Plewa (who comes to us from the Copernicus Center in Warsaw, via the Max Planck Institute for Theoretical Astrophysics in Garching bei München, Germany); Plewa is expected to start in February 2001.

8.2 Faculty

The faculty additions resulting from the creation of the FLASH Center at Chicago now include F. Cattaneo (assist. professor, Dept. of Mathematics), R. Kirby (L.E. Dickson Instructor, Depts. of Computer Science and Mathematics), A. Kiselev (assist. professor, Dept. of Mathematics), L. Ryzhik (L.E. Dickson Instructor, Dept. of Mathematics), R. Stevens (professor, Dept. of

Computer Science). The build-up of computational science at Chicago also led to the arrival of Ridgway Scott from Houston; Scott is now a member of the FLASH Center, and co-director of the Computations Institute.

9 Education and Comp. Science at Chicago

The University of Chicago at the highest levels is involved in a broad study of the role of computation in education and research. In his recent annual report on The State of the University, the Provost listed computation as his first focus area under new initiatives. A committee, chaired by Deputy Provost Robert Zimmer with Rick Stevens as Associate Chair, has been formed and will report to the administration soon, probably in the winter quarter. Our new administration has indicated a willingness to entertain bold proposals in this area. This is a cause for considerable optimism about the future of computation here.

In the meanwhile, we have not waited for committee reports, and simply proceeded in two major directions: first, an enhancement of teaching activities in the computational sciences, and second, the creation of a new research institute which serves as a "home" for computationally-related research activities, and directly bridges such activities at the University and at Argonne National Laboratory.

9.1 Students

There are now a total of 12 graduate students actively working on the FLASH Center problems from four departments (Astrophysics, Computer Science, Mathematics, and Physics); five have graduated this past year, and two are about to graduate.

Three graduate students are currently working on the astrophysics portion of the Center's research: J. Dursi (supervisor R. Rosner), A. Mignone (supervisor R. Rosner), and F. Peng (supervisor J. Truran). Two of the students (Dursi and Mignone) are also closely associated with the Code Group. The Computer Science students are focusing on visualization (M. Papka; supervisor R. Stevens) and exploration of tools for unstructured computations (E. Gomez, supervisor R. Scott). The mathematicians are focusing on flame theory (A. Oberman, supervisor P. Constantin) and level set methods and mixing (A. Drageanescu, supervisor T. Dupont); the students in Physics are working on interface instabilities and mixing (A. Alexakis, supervisor R. Rosner; and M. Medved, supervisor H. Jaeger), code physics (A. Caceres, no supervisor as yet), flame modeling (O. Ruchayskiy, supervisor R. Rosner), and multiply-diffusive instabilities (J. Curtis, supervisor D. Grier).

Five former students completed their thesis research and have received their PhD degrees during this past year: Y.-N. Young (supervisor R. Rosner), on *Mixing Instabilities in Astrophysics*, now a postdoctoral fellow in the Applied Mathematics Dept. of Northwestern University; Yingjie Liu (supervisor T. Dupont), on *Symmetric Error Estimates for Moving Mesh Finite Element Methods*, now in Jim Glimm's group at Stony Brook; R. Loy (supervisor J. Flaherty, RPI), now a postdoctoral fellow in MCS/ANL, and a member of the FLASH Center; M. Zingale (supervisor J. Truran), on *Helium Detonations on Neutron Stars*, now a postdoctoral fellow within the FLASH Center here in Chicago; and S. Zhan (supervisor D.Q. Lamb), on *Thermal Structure and Thermonuclear Flashes in Accreting Neutron Star Envelopes*, now in private industry.

9.2 Teaching

The Computer Science Department has substantially increased its course offerings relevant to FLASH Center activities, and FLASH Center-related scientists are teaching in its program. Last spring R. Stevens taught Parallel Computer Architecture (CS324-01), which provided graduate level discussion of advanced parallel machines and the techniques for measuring and modeling performance of parallel computers. Courses taught this autumn include

- Beowulf (CS103-01; H. Tufo, instructor)
- Computer Architecture (CS322-01; R. Stevens)
- Scalable Internet Services (CS347-01; I. Foster)

The Department has also generally increased its course offerings in areas of direct relevance to our Center, such as courses on Networks and Distributed Systems (D. Beazley) and Matrix Computation (CS378; Y. Amit). The department will also offer a new course sequence starting in the winter for computer graphics and visualization that is directly relevant to the development of visualization tools needed by the Center. Also, this winter and spring there will be a two course sequence on High Performance Computing on the Internet (CS329 & 339; I. Foster); a Winter quarter course on parallel scientific computing (CS340; R. Scott); and a Spring quarter course on Computational Fluid Dynamics (CS384; T. Dupont). In addition, the Department of Mathematics is expanding its offerings in applied mathematics, with courses on computations taught by F. Cattaneo.

9.3 Computation Institute

The University of Chicago and Argonne National Laboratory jointly founded a new institute, the Computation Institute (CI) in the autumn of 1999. Co-directed by R. Scott and R. Stevens, the aim of this institute is to play a major role in facilitating the interactions between computer scientists, applied mathematicians, and applications scientists at both the University and at Argonne.

The Institute will focus on leveraging the more obvious and well-developed computational science activities that currently exist such as the FLASH Center and computational astrophysics to emerging areas like computational biology and computational archaeology.

The CI also plays a major role in catalyzing the development of a formal computational science curriculum at the University in the next year or so. This initiative had among its multiple roots the Computational and Applied Mathematics program (CAMP; http://www.math.uchicago.edu/camp) at Chicago as well as the extensive interactions between Chicago computationally-oriented scientists and scientists within the Argonne MCS. The current plan is to establish a Committee on Computational Science. At The University of Chicago, this is called a "Committee with a capital C" and is almost like a department. This is to be a program that will grant Ph.D.'s in Computational Science. The details of this program, which have been worked out by members of the CI, including several FLASH Center members, include degree requirements, course descriptions, and Committee membership; this plan is to be submitted for approval to the appropriate University faculty oversight committees by Winter quarter.

What makes the CI unique among the existing institutes at the University is its relationship with Argonne National Laboratory. The other institutes are all "creatures" of the University alone, while the CI is truly a joint enterprise of the two institutions. The Institute is currently in the midst of organizing and fundraising. A first retreat was held at the University's Gleacher Center in late September 1999, and attracted 70 senior scientists from the University and from Argonne. Issues that the Institute leadership is addressing in the near term include: the types of activities supported by the CI, the nature and number of appointments to the CI, and the issue of space and infrastructure resources. We believe that the CI is a very important outcome, not only because it was strongly influenced by the success of the FLASH center as a computational science project, but also as an example of University and Argonne cooperative activity. More information can be obtained at the CI's web site http://www-fp.mcs.anl.gov/ci.

10 Infrastructure

10.1 Space

In the past year, only very modest changes were made to our office space. The major change entailed the re-working of a large room previously used for small computer classes as a workshop for visualization and for hosting an AccessGrid node that will provide always-on collaborative video and audio capabilities linking Chicago and Argonne.

10.2 High Performance Storage System (HPSS)

We have been struggling for some time to bring IBM's HPSS to campus and ANL; at this point, the paperwork that we needed to do has been completed, the software has been installed (!), and discussions are proceeding between the hardware/software group taking care of the HPSS and the Code/Astro groups within the FLASH Center on how best to configure the system for use. We plan to be using HPSS on a routine basis within the next few months; a test of the HPSS by computational astrophysicists moving data from Chicago to the HPSS has already taken place, and tests of moving data from the DP Labs to the HPSS will be carried out shortly.

10.3 Center Web Site

Our Center web site (http://flash.uchicago.edu/) is continuously updated, including a gallery of computational results, full descriptions of the activities of the various research groups and teams within the Center, and the documentation for the *Flash* code.

11 Bibliography

References

[1] Brackbill, J. & Barnes, D.C. 1980, J. Comp. Phys., 35, 426.

[2] Brown, E.F., & Bildsten, L. 1998, *The ocean and crust of a rapidly accreting neutron star: Implications for magnetic field evolution and thermonuclear flashes*, ApJ, 496, 915.

[3] Calder, A., Curtis, B.C., Dursi, L.J., Fryxell, B., Henry, G., MacNeice, P., Olson, K., Ricker, P., Rosner, R., Timmes, F.X., Tufo, H.M., Truran, J.W., & Zingale, M. 2000, *High-Performance Reactive Fluid Flow Simulations Using Adaptive Mesh Refinement on Thousands of Processors*, Supercomputing 2000, in press.

[4] Champagne, A. E., & Wiescher, M. 1992, *Explosive hydrogen burning*, Ann. Rev. Nucl. Part. Sci., 42, 39.

[5] Chan, A., Gropp, W., & Lusk, E. 2000, *Scalable Log Files for Parallel Program Trace Data*, ANL preprint

[6] Colella, P., & Woodward, P.R. 1984, JCP, 54, 174-201.

[7] Constantin, P., Kiselev, A., Oberman, A., & Ryzhik, L. 2000, *Bulk Burning Rate in Passive - Reactive Diffusion*, Arch. Rational Mechanics, **154**, 53-91.

[8] Couchman, H.M.P. 1991, ApJ, 368, 23.

[9] Fryxell, B., Müller, E., & Arnett, W.D. 1989, *Hydrodynamics and Nuclear Burning*, Max–Planck–Institut für Astrophysik Report 449.

[10] Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J., & Tufo, H. 2000, *FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes*, ApJ, in press.

[11] Glasner, S.A., Livne, E., & Truran, J.W. 1997, ApJ, 475, 754.

[12] Gomez, E., & Scott, L.R. 1998, *Overlapping and Short-Cutting Techniques in Loosely Synchronous Irregular Problems*, LNCS No. 1457, Ferreira, Rolim, Simon & Teng, eds. Springer-Verlag) pp. 116-127.

[13] Gropp, W., Lusk, E.R., & Skjellum, A. 1999, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, 2nd edition (Cambridge: MIT Press).

[14] James, R.A. 1977, JCP, 25, 71.

[15] Karonis, N.T., de Supinski, B.R., Foster, I., Gropp, W., Lusk, E., & Bresnahan, J. 2000, *Exploiting Hierarchy in Parallel Computer Networks to Optimize Collective Operation Performance*, Proceedings of IPDPS2000, to appear.

[16] Kercek, A., Hillebrandt, W., & Truran, J.W. 1998, A&A, 337, 379.

[17] Khokhlov, A.M. 1995, ApJ, 449, 695.

[18] Khokhlov, A.M., Oran, E.S., & Wheeler, J.C. 1997, ApJ, 478, 678.

[19] Kippenhahn, R. & Thomas, H.-C. 1978, A&A, 63, 625.

[20] Kiselev, A., & Ryzhik, L. 2000, *Enhancement of the Traveling Front Speeds in Reaction-Diffusion Equations with Advection*, Annales l'Institut Henri Poincare, submitted.

[21] Koike, O., Hashimoto, M., Arai, K., & Wanajo, S. 1999, *Rapid proton capture on accreting neutron stars—effects of uncertainty in the nuclear process*, AAP, 342, 464-73.

[22] Litwin, C., Rosner, R., & Lamb, D.Q. 1999a, MNRAS, 310, 324-30; Litwin, C., Rosner, R., & Lamb, D.Q. 1999b, Plasma Phys. Control. Fusion, 41, B343-B350.

[23] Litwin, C., Brown, E.F., & Rosner, R. 2000, *Ballooning instability in polar caps of accreting neutron stars*, ApJ, in press.

[24] Livio, M. & Truran, J.W. 1994, ApJ, 425, 797.

[25] Livio, M. & Truran, J.W. 1990, in *Nonlinear Astrophysical Fluid Dynamics*, Buchler, J.-R. & Gottesman, S.T. (Eds.), New York Academy of Sciences, 617, 126.

[26] Marder, B. 1987, J. Comp. Phys., 68, 48.

[27] Niemeyer, J. & Woosley, S.A. 1997, ApJ, 475, 740.

[28] PARAMESH 1999, see web site at http://esdcd.gsfc.nasa.gov/ESS/macneice/paramesh/paramesh.html

[29] Plewa, T. 2000, private communication.

[30] Powell, K.G., Roe, P.L., Linde, T.J., Gombosi, T.I., & DeZeeuw, D.L. 1999, J.Comp.Phys., 154, 284.

[31] Rembges, J.F., Liebendörfer, M., Rauscher, T., Thielemann, F.K., & Schatz, H. 1998, *Hydrogen consumption in x-ray bursts*, in *Proceedings of the 9th workshop on Nuclear Astrophysics, eds. W. Hillebrandt & E. Muller (Garching bei Munchen: Max-Planck-Institut fur Astrophysik)*, p. 71.

[32] Schatz, H., Aprahamian, A., Goerres, J., Wiescher, M., Rauscher, T., Rembges, J.F., Thielemann, F.K., Pfeiffer, B., Moller, P., Kratz, K.L., Herndl, H., Brown, B.A., & Rebel, H. 1998, *rp-process nucleosynthesis at extreme temperature and density conditions*, Phys. Rep., 294(4), 167.

[33] Rubini, F., & Taverne, V. 1999, private communication.

[34] Schatz, H., Bildsten, L., Cumming, A., & Wiescher, M. 1999, *The rapid proton process ashes from stable nuclear burning on an accreting neutron star*, ApJ, 524, 1014-29.

[35] Sedov, L.I. 1959, *Similarity and Dimensional Methods in Mechanics* (New York: Academic)

[36] Sod, G. 1978, JCP, 27, 1.

[37] SUMAA3d 1999, see web site at http://www-unix.mcs.anl.gov/sumaa3d

[38] Timmes, F.X. 1999, ApJ Suppl., 124, 241.

[39] Timmes, F.X. 2000, ApJ, 528, 913.

[40] Timmes, F.X., & Arnett, W.D. 1999, ApJS, 125, 277.

[41] Timmes, F.X., Hoffman, R.D., & Woosley, S.E. 2000, ApJ Suppl., 129, 377.

[42] Timmes, F.X., & Swesty, F.D. 2000, ApJS, 126, 501.

[43] Timmes, F.X., & Woosley, S.E. 1992, ApJ, 396, 649.

[44] Timmes, F.X., Zingale, M., Olson, K., Fryxell, B., Ricker, P., Calder, A.C., Dursi, L.J., Truran, J.W., & Rosner, R. 2000, *On the Cellular Structure of Carbon Detonations*, ApJ, in press.

[45] Timmes, F.X., Zingale, M., Dursi, L.J., Fryxell, B., Ricker, P., Olson, K., Calder, A.C., Tufo, H.M., MacNeice, P., Truran, J.W., & Rosner, R. 2000, *Similarities and Differences Between the 2D and 3D Cellular Structure of Carbon Detonations*, to be submitted to ApJ.

[46] Truran, J.W. 1982, in *Essays in Nuclear Astrophysics*, Barnes, C.A., Clayton, D.D., & Schramm, D.N. (Eds.), Cambridge University Press, p. 467.

[47] Tufo, H.M., & Fischer, P.F. 1999, in *Fast Parallel Direct Solvers for Coarse Grid Problems*, J. Par. & Dist. Comput., in press.

[48] van Leer, B. 1979, J. Comp. Phys., 32, 101.

[49] Van Wormer, L., Görres, J., Iliadis, C., Wiescher, M., & Thielemann, F.K. 1994, *Reaction rates and reaction sequences in the rp-process*, ApJ, 432, 326.

[50] Weaver, T.A., & Woosley, S.E. 1993, *Nucleosynthesis in massive stars and the 12C(alpha,gamma)16O reaction rate*, Phys. Rep., 227, 65.

[51] Woodward, P., & Colella, P. 1984, JCP, 54, 115.

[52] Wu, E.E., Bolmarcich, A., Snir, M., Wootton, D., Parpia, F., Chan, A., Lusk, E., Gropp, W. 2000, *From Trace Generation to Visualization: A Performance Framework for Distributed Parallel Systems*, Proceedings of SC'00 (November, 2000),to appear (refereed conference proceedings).

[53] Young, Y.-N., Tufo, H., Rosner, R., & Dubey, A. 1999, JFM, submitted.

[54] Vladimirova, N., Malagoli, A., Cattaneo, F., & Ruchayskiy, O. 2000, PRL, submitted.

[55] Zingale, M., Timmes, F.X., Fryxell, B., Lamb, D.Q., Olson, K., Calder, A.C., Dursi, L.J., Ricker, P., Rosner, R., MacNeice, P., & Tufo, H. 2000, ApJS, in press.

12 Appendix: Center Members and Affiliates (Oct. 1, 2000)

The following table includes all scientists and support staff who receive full or partial support from the FLASH Center.

Name	Position	Center Affiliation	Institutional Unit(s)	Institution
Robert Rosner	Faculty	Director Astro/Code/V&P	A&A/Physics/EFI	UChicago
Alex Alexakis	Graduate student	V&P	Physics	UChicago
Guillaume Bal	Research scientist	V&P	Math	UChicago
Edward Brown	Research scientist	Astro	A&A/EFI	UChicago
Alvero Caceres	Graduate student	Code	Physics	UChicago
Alan Calder	Research scientist	Astro/Code	A&A	UChicago
Fausto Cattaneo	Faculty	V&P	Math	UChicago
Anthony Chan	Comput. staff	CS	MCS	UChicago/ANL
Carrie Clark	Admin. staff	-	-	UChicago
Terry Clark	Research scientist	CS	CS/CI	UChicago
Peter Constantin	Faculty	V&P	Math	UChicago
Lois Curfman-McInnes	Research scientist	CS	MCS	ANL
Jennifer Curtis	Graduate student	V&P	Physics	UChicago
Damien Dawson	Graduate student	V&P	Physics	UIUC
Andrei Draganescu	Graduate student	V&P	Math	UChicago
Todd F. Dupont	Faculty	V&P	CS/Math/JFI	UChicago
Jonathan Dursi	Graduate student	Astro/Code	A&A	UChicago
Joseph E. Flaherty	Faculty	CS	CS	RPI
Paul Fischer	Senior Researcher	CS/V&P	MCS	ANL
Ian T. Foster	Senior researcher	CS	MCS	ANL
	Faculty		CS	UChicago
Lori A. Freitag	Senior Researcher	CS	MCS	ANL
Bruce Fryxell	Senior researcher	Astro/Code/V&P	EFI	UChicago
Ernesto Gomez	Graduate student	CS	CS	UChicago
William D. Gropp	Senior researcher	CS	MCS	ANL
Randy Hudson	Comput. staff	CS	MCS	UChicago/ANL
Christophe Josserand	Research scientist	V&P	JFI	UChicago
Leo Kadanoff	Faculty	V&P	Phys./Math/EFI/JFI	UChicago
Alexander Kiselev	Faculty	V&P	Math	UChicago
Mila Kuntu	Admin. staff	-	-	UChicago
Don Q. Lamb	Faculty	Astro/EFI	A&A	UChicago
Timur Linde	Research scientist	Code/V&P	A&A	UChicago
Christof Litwin	Senior researcher	Astro/V&P	A&A	UChicago
Ray Loy	Research scientist	CS	MCS	ANL

Table continued on next page.

Table continued from previous page.

Name	Position	Center Affiliation	Institutional Unit(s)	Institution
Ewing L. Lusk	Senior researcher	CS	MCS	ANL
Ruben Krasnopolsky	Research scientist	Astro/V&P	A&A	UChicago
Andrea Malagoli	Senior researcher	V&P	A&A	UChicago
Samuel Meder	Staff scientist	CS	CS	UChicago
Milica Medved	Graduate student	V&P	Physics	UChicago
Andrea Mignone	Graduate student	Astro/Code	A&A	UChicago
Tom Morgan	Admin. staff	CS	MCS	ANL
Jens C. Niemeyer	Research scientist	Astro	-	MPI Theo. Astro.
Adam Oberman	Graduate student	V&P	Math	UChicago
Kevin Olson	Senior researcher	Astro/Code	EFI	UChicago
Michael Papka	Graduate student/Staff scientist	CS	CS/MCS	UChicago/ANL
Fang Peng	Graduate student	Astro	A&A	UChicago
Ray Pierehumbert	Faculty	V&P	Geosci.	UChicago
Paul E. Plassmann	Faculty	CS	CS	PSU
J.-P. Remacle	Research scientist	CS	Applied Math	RPI
Paul Ricker	Research scientist	Astro/Code	A&A	UChicago
Katherine Riley	Comput. staff	Code	-	UChicago
Oleg Ruchayksiy	Graduate student	V&P	Physics	UChicago
Lenya Ryzhik	Faculty	V&P	Math	UChicago
Ridgway Scott	Faculty	CS	CS/Math	UChicago
Mark S. Shephard	Faculty	CS	CS	RPI
Andrew Siegel	Comput. staff	Code	-	UChicago
Barry F. Smith	Senior researcher	CS	MCS	ANL
Rick Stevens	Senior researcher	CS	MCS	ANL
	Faculty		CS	UChicago
Frank X. Timmes	Research scientist	Astro/Code	A&A	UChicago
James W. Truran	Faculty	Astro	A&A/EFI	UChicago
Henry Tufo	Research scientist	CS/Code/V&P	CS/MCS	UChicago/ANL
Natasha Vladimirova	Research scientist	Code/V&P	A&A	UChicago
Nektarios Vlahakis	Research scientist	Astro/V&P	A&A	UChicago
Greg Weirs	Research scientist	Code/V&P	A&A	UChicago
Kevin Young	Staff scientist	Code	A&A	UChicago
Yuan-nan Young	Postdoctoral fellow	V&P	Applied Math	NWU
Michael Zingale	Graduate student	Astro/Code	A&A	UChicago

Table acronym definitions:

- Center Affiliation: Astro: Astrophysics group; CS: Computer Science group; Code: Flash Code group; V&P: Validation/Basic Science group.
- Institutional Unit: A&A: Dept. of Astronomy & Astrophysics; CS: Dept. of Computer Science; EFI: Enrico Fermi Institute; Geosci.: Dept. of

Geophysical Sciences; JFI: James Frank Institute; MCS: Mathematics and Computer Science Division; Phys.: Dept. of Physics

- Institution: ANL: Argonne National Laboratory; NWU: Northwestern University; PSU: Pennsylvania State University; RPI: Rennselear Polytechnic Institute; UChicago: The University of Chicago; UIUC: University of Illinois at Urbana/Champaign

13 Publications

A continuously updated publications list is provided at our website, at http://flash.uchicago.edu/publications/ . The publication list, as of Oct. 1, 2000, is given below.

1. B. Ayata and T. F. Dupont, *Convergence of a Step-Doubling Galerkin Method for Parabolic Problems*, to be submitted.

2. R. Bank, T. F. Dupont, S. Garcia, Y. Liu, and R. Santos, *Symmetric Error Estimates for Moving Mesh Mixed Methods for Advection Diffusion Equations*, SIAM Jour Numer Anal (2000), submitted.

3. K. E. Belle, C. E. Woodward, A. Evans, S. Eyres, R. D. Gehrz, M. Schuster, M. A. Greenhouse, J. Krautter, S. G. Starrfield, and J. Truran, *Nicmos Imagery of Nova Shells*, American Astronomical Society Meeting, vol. 194, May 1999, p. 8611.

4. A. C. Calder, B. Curtis, L. J. Dursi, B. Fryxell, G. Henry, P. MacNeice, K. Olson, P. Ricker, R. Rosner, F. X. Timmes, H. M. Tufo, J. W. Truran, and M. Zingale, *High-Performance Reactive Fluid Flow Simulations using Adaptive Mesh Refinement on Thousands of Processors*, SC 2000 (2000), Gordon Bell submission.

5. A. C. Calder, B. Fryxell, R. Rosner, J. Kane, B. A. Remington, L. J. Dursi, K. Olson, P. M. Ricker, F. X. Timmes, M. Zingale, P. MacNeice, and H. Tufo, *Flash Simulations of Multi-Layer Targets*, American Astronomical Society Meeting, vol. 196, May 2000, p. 2203.

6. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets.*, Journal of Network and Computer Applications (2000).

7. T. F. Dupont and Y. Liu, *Symmetric Error Estimates for Moving Mesh Galerkin Methods for Advection Diffusion Equations*, SIAM Jour. Numer Anal (2000), submitted.

8. J. Dursi, J. Niemeyer, A. Calder, B. Fryxell, D. Lamb, K. Olson, P. Ricker, R. Rosner, F. Timmes, H. Tufo, and M. Zingale, *Compressed Reactive Turbulence and Supernovae ia Recollapse using the Flash Code*, American Astronomical Society Meeting, vol. 195, December 1999, p. 4202.

9. P. F. Fischer, N. I. Miller, and H. M. Tufo, *An Overlapping Schwarz Method for Spectral Element Simulation of Three-Dimensional Incompressible Flows*, Parallel Solution of Partial Differential Equations (2000).

10. B. Fryxell, K. Olson, P. M. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. M. Tufo, *Flash: a Multidimensional Hydrodynamics Program for Modeling Astrophysical Fluids and Thermonuclear Flashes*, ApJ (2000), in press.

11. B. Fryxell, F. X. Timmes, M. Zingale, L. J. Dursi, P. Ricker, K. Olson, A. C. Calder, H. Tufo, P. MacNeice, J. W. Truran, and R. Rosner, *The Cellular Structure of Carbon Detonations*, American Astronomical Society Meeting, vol. 196, May 2000, p. 3902.

12. B. Fryxell, M. Zingale, F. X. Timmes, D. Q. Lamb, K. Olson, A. C. Calder, L. J. Dursi, P. Ricker, R. Rosner, J. W. Truran, P. MacNeice, and H. Tufo, *Numerical Simulations of Thermonuclear Flashes on Neutron Stars*, Nuclear Physics A, in press (2000).

13. B. Fryxell, M. Zingale, F. X. Timmes, K. Olson, D. Lamb, A. Calder, J. Dursi, P. Ricker, R. Rosner, and H. Tufo, *Helium Burning on Neutron Stars: 2-Dimensional Results*, American Astronomical Society Meeting, vol. 195, December 1999, p. 4204.

14. R. D. Gehrz, J. W. Truran, R. E. Williams, and S. Starrfield, *Nucleosynthesis in Classical Novae and its Contribution to the Interstellar Medium*, PASP **110** (1998), 3.

15. J. Kane, D. Arnett, B. A. Remington, S. G. Glendinning, G. Bazan, R. P. Drake, and B. A. Fryxell, *Supernova Experiments on the Nova Laser*, ApJS **127** (2000), 365.

16. J. Kane, D. Arnett, B. A. Remington, S. G. Glendinning, G. Bazán, E. Muller, B. A. Fryxell, and R. Teyssier, *Two-Dimensional versus Three-Dimensional Supernova Hydrodynamic Instability Growth*, ApJ **528** (2000), 989.

17. A. Kercek, W. Hillebrandt, and J. W. Truran, *Two- and Three-Dimensional Simulations of the Thermonuclear Runaway in an Accreted Atmosphere of a C+O White Dwarf*, LNP Vol. 287: Proceedings of the 9th workshop on Nuclaear Astrophysics, 1998, p. 56.

18. A. Kercek, W. Hillebrandt, and J. W. Truran, *Two- and Three-Dimensional Simulations of Classical Novae*, Astronomische Gesellschaft Meeting Abstracts, Abstracts of Contributed Talks and Posters presented at the Annual Scientific Meeting of the Astronomische Gesellschaft at Heidelberg, September 14–19, 1998, talk #C10 **14** (1998), 10.

19. A. Kercek, W. Hillebrandt, and J. W. Truran, *Two-Dimensional Simulations of the Thermonuclear Runaway in an Accreted Atmosphere of a C+O White Dwarf*, A&A **337** (1998), 379.

20. A. Kercek, W. Hillebrandt, and J. W. Truran, *Three-Dimensional Simulations of Classical Novae*, A&A **345** (1999), 831.

21. D. Q. Lamb, *Some Startling Discoveries about X-Ray Bursts*, ApJS **127** (2000), 395.

22. T. Linde and A. Malagoli, *On Local Div B Control in Ideal MHD Simulations*, JCP (2000), submitted.

23. C. Litwin and R. Rosner, *Plasmoid Impacts on Neutron Stars and Highest Energy Cosmic Rays*, Phys. Rev. Let. (2000), submitted.

24. C. Litwin and R. Rosner, *Relativistic Space-Charge Limited Bipolar Flow*, Phys. Rev. E **58** (2000), 1163.

25. C. Litwin, R. Rosner, and D. Q. Lamb, *On Accretion Flow Penetration of Magnetospheres*, MNRAS **310** (1999), 324.

26. C. Litwin, R. Rosner, and D. Q. Lamb, *On Stream Accretion onto Magnetized Compact Objects*, American Astronomical Society Meeting, vol. 194, May 1999, pp. 2909.

27. Christof Litwin, Edward F. Brown, and R. Rosner, *Ballooning Instability in Polar Caps of Accreting Neutron Stars*, ApJ (2000), accepted.

28. E. Marietta, A. Burrows, and B. Fryxell, *Type Ia Supernova Explosions in Binary Systems: the Impact on the Secondary Star and its Consequences*, ApJS **128** (2000), 615.

29. J. C. Niemeyer and J. W. Truran (eds.), *Type Ia Supernovae: Theory and Cosmology*, 2000.

30. K. M. Olson, P. MacNeice, B. Fryxell, P. Ricker, F. X. Timmes, and M. Zingale, *Paramesh: a Parallel, Adaptive Mesh Refinement Toolkit and Performance of the ASCI/Flash Code*, American Astronomical Society Meeting, vol. 195, December 1999, p. 4203.

31. P. M. Ricker, S. Dodelson, and D. Q. Lamb, *Cosmos: a Hybrid N-Body /Hydrodynamics Code for Cosmological Problems*, ApJ **536** (2000), 122.

32. P. M. Ricker, B. Fryxell, K. Olson, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, and H. Tufo, *Flash: a Multidimensional Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes*, American Astronomical Society Meeting, vol. 195, December 1999, p. 4205.

33. R. Rosner, A. C. Calder, L. J. Dursi, B. Fryxell, D. Q. Lamb, J. C. Niemeyer, K. Olson, P. M. Ricker, F. X. Timmes, J. W. Truran, H. M. Tufo, Y. N. Young, M. Zingale, E. Lusk, and R. Stevens, *Astrophysical Thermonuclear Flashes*, Comp. in Sci. and Eng. **2** (2000), no. 5.

34. A. A. Sonzogni, K. E. Rehm, I. Ahmad, F. Borasi, D. L. Bowers, F. Brumwell, J. Caggiano, C. N. Davids, J. P. Greene, B. Harss, A. Heinz, D. Henderson, R. V. F. Janssens, C. L. Jiang, G. McMichael, J. Nolen, R. C. Pardo, M. Paul, J. P. Schiffer, R. E. Segel, D. Seweryniak, R. H. Siemssen, J. W. Truran, J. Uusitalo, I. Wiedenhoever, and B. Zabransky, *The $^{44}Ti(\alpha,p)$ Reaction and its Implication on the ^{44}Ti Yield*, Physical Review Letters **84** (2000), 1651.

35. S. Starrfield, G. J. Schwarz, J. W. Truran, and W. M. Sparks, *Hydrodynamic Studies of the Evolution of the Extraordinary Nova LMC 1991*, American Astronomical Society Meeting, vol. 193, December 1998, p. 10106.

36. S. Starrfield, G. J. Schwarz, J. W. Truran, and W. M. Sparks, *The Effects of Metallicity on Hydrodynamic Studies of the Nova Outburst*, American Astronomical Society Meeting, vol. 194, May 1999, p. 8609.

37. S. Starrfield, W. M. Sparks, J. W. Truran, and M. C. Wiescher, *The Effects of New Nuclear Reaction Rates and Opacities on Hydrodynamic Simulations of the Nova Outburst*, ApJS **127** (2000), 485.

38. S. Starrfield, J. W. Truran, and W. M. Sparks, *The Nova Outburst: Thermonuclear Runaways on Degenerate Dwarfs*, New Astronomy Review **44** (2000), 81.

39. S. Starrfield, J. W. Truran, M. C. Wiescher, and W. M. Sparks, *Evolutionary Sequences for Nova V1974 Cygni using New Nuclear Reaction Rates and Opacities*, MNRAS **296** (1998), 502.

40. D. Swesty, E. Y. M. Wang, and A. C. Calder, *Numerical Models of Binary Neutron Star System Mergers. I.: Numerical Methods and Equilibrium Data for Newtonian Models*, ApJ (2000), accepted.

41. F. X. Timmes, *Integration of Nuclear Reaction Networks for Stellar Hydrodynamics*, ApJS **124** (1999), 241.

42. F. X. Timmes, *Physical Properties of Laminar Helium Deflagrations*, ApJ **528** (2000), 913.

43. F. X. Timmes and D. Arnett, *The Accuracy, Consistency, and Speed of Five Equations of State for Stellar Hydrodynamics*, ApJS **125** (1999), 277.

44. F. X. Timmes, R. Hoffman, and S. E. Woosley, *An Inexpensive Nuclear Energy Generation Network for Stellar Hydrodynamics*, ApJ (2000), in press.

45. F. X. Timmes and J. C. Niemeyer, *Regimes of Helium Burning*, ApJ **537** (2000), 993.

46. F. X. Timmes and F. D. Swesty, *The Accuracy, Consistency, and Speed of an Electron-Positron Equation of State Based on Table Interpolation of the Helmholtz Free Energy*, ApJS **126** (2000), 501.

47. F. X. Timmes, M. Zingale, K. Olson, B. Fryxell, P. M. Ricker, A. C. Calder, L. J. Dursi, H. M. Tufo, P. MacNeice, J. W. Truran, and R. Rosner, *On the Cellular Structure of Carbon Detonations*, ApJ (2000), in press.

48. F. X. Timmes, M. Zingale, K. Olson, P. Ricker, B. Fryxell, A. C. Calder, L. J. Dursi, J. W. Truran, and R. Rosner, *On the Cellular Structure of 3-Dimensional Carbon Detonations*, ApJ (2000), in preparation.

49. J. W. Truran, *The Early Stages of a Nova Outburst*, Stellar Evolution, Stellar Explosions and Galactic Chemical Evolution, 1998, p. 425.

50. J. W. Truran, *Nova Explosions: Abundances and Light Curves*, LNP Vol. 287: Proceedings of the 9th workshop on Nuclear Astrophysics, 1998, p. 107.

51. J. W. Truran and J. J. Cowan, *The Helium R-Process and Neutron-Capture Element Abundances in Low Metallicity Stars*, American Astronomical Society Meeting, vol. 196, May 2000, p. 0513.

52. Y. Tuchman and J. W. Truran, *Composition Influence upon the Core Mass-Luminosity Relation and Possible Implications for the Extinction Timescales of Classical Novae*, ApJ **503** (1998), 381.

53. H. M. Tufo and P. F. Fischer, *Fast Parallel Direct Solvers for Coarse Grid Problems*, J. Par. & Dist. Computing (2000), in press.

54. D. Uzdensky, A. Konigl, and C. Litwin, *Force-Free Models of Magnetically Linked Star-Disk Systems*, ApJ (2000), to be submitted.

55. S. Vainshtein, Z. Mikic, R. Rosner, and J. Linker, *Evidence for Topological Nonequilibrium in Magnetic Configurations*, Phys. Rev. E, **62** (2000), 1245-51.

56. A. Venkatesan, A. V. Olinto, and J. W. Truran, *Neutron Stars and Black Holes as Machos*, ApJ **516** (1999), 863.

57. Y. N. Young, and R. Rosner, *Linear and Weakly Non-linear Analysis of Doubly Diffusive Vertical Slot Convection*, Phys. Rev. E, **57** (1998), 5554-63.

58. Y. N. Young, and R. Rosner, *Numerical Simulation of Double-Diffusive Convection in a Rectangular Box*, Phys. Rev. E, **61(3)** (2000), 2676-2694.

59. Y. N. Young, H. Tufo, A. Dubey, and R. Rosner, *Miscible Rayleigh-Taylor Instability: 2D versus 3D*, American Astronomical Society Meeting, vol. 196, May 2000, p. 2206.

60. Y. N. Young, H. M. Tufo, A. Dubey, and R. Rosner, *On the Miscible Rayleigh-Taylor Instability: Mixing Layer Width Scaling in 2- and 3-D*, JFM (2000), in press.

61. Y. N. Young and S. Wunsch, *Universality of Scalar Statistics in Stably Stratified Turbulence*, Phys. Rev. Let. (2000), submitted.

62. M. Zingale, F. X. Timmes, A. Calder, J. Dursi, B. Fryxell, D. Lamb, K. Olson, P. Ricker, R. Rosner, J. Truran, and H. Tufo, *Direct Simulations of Thermonuclear Flames with the Flash Code*, American Astronomical Society Meeting, vol. 195, December 1999, p. 4201.

63. M. Zingale, F. X. Timmes, B. Fryxell, D. Q. Lamb, K. Olson, A. C. Calder, L. J. Dursi, P. Ricker, R. Rosner, J. W. Truran, P. MacNeice, and H. Tufo, *Helium Detonations on Neutron Stars*, American Astronomical Society Meeting, vol. 196, May 2000, p. 1703.

64. M. Zingale, F. X. Timmes, B. Fryxell, D. Q. Lamb, K. Olson, A. C. Calder, L. J. Dursi, P. M. Ricker, R. Rosner, P. MacNeice, and H. M. Tufo, *Helium Detonations on Neutron Stars*, ApJS (2000), in press.

65. M. Zingale, F. X. Timmes, B. Fryxell, D. Q. Lamb, K. Olson, P. Ricker, A. C. Calder, L. J. Dursi, R. Rosner, and J. W. Truran, *2-Dimensional Helium Detonations on the Surface of Neutron Stars*, Rossi2000: Astrophysics with the Rossi X-ray Timing Explorer, 2000, p. E67.

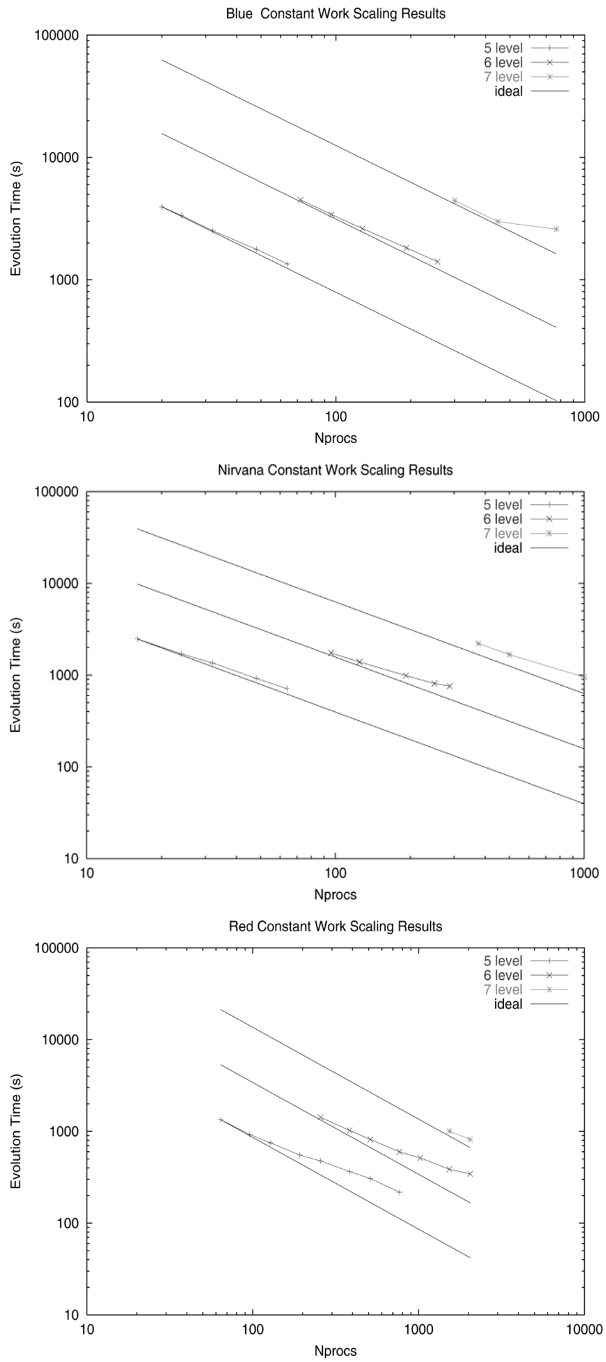


Figure 1: These three panels illustrate the scaling performance of *Flash-1.61* on the three accessible ASCI platforms (from top) Blue Pacific (LLNL), Nirvana (LANL), and Red (SNL).

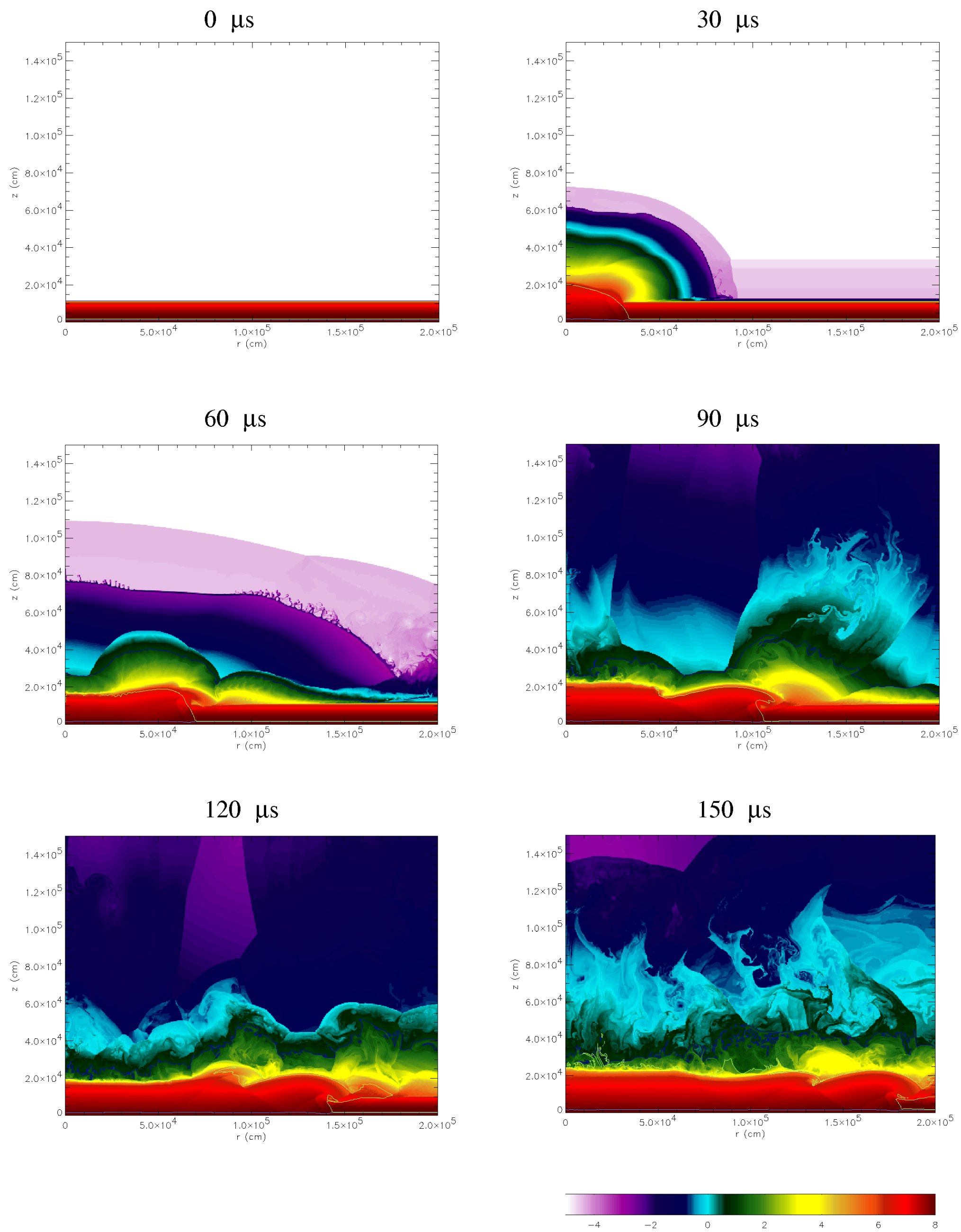


Figure 2: This figure displays the density structure of the neutron star envelope at 6 times over the course of an X-ray outburst. The green curve marks a helium abundance of 0.9; below this line, burning has begun to deplete the helium. The dark blue curve marks 10 g cm^{-3}, and indicates how much the neutron star atmosphere has been distorted by the explosion.

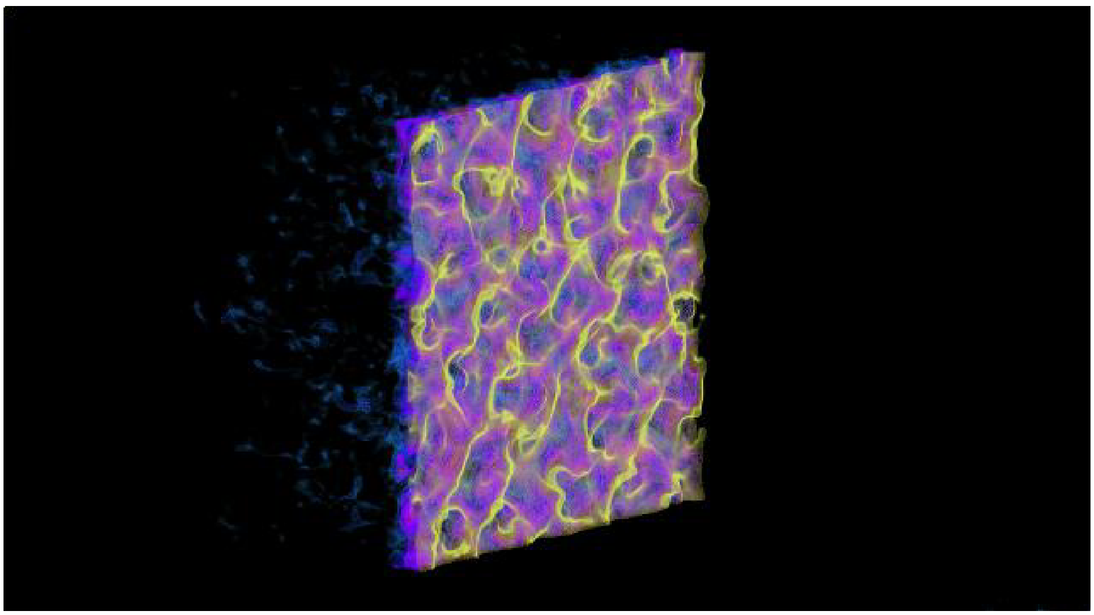


Figure 3: This figure shows the distribution of silicon ashes formed behind a detonation front inside of a Type 1a supernova explosion. Note the cellular structure, and the evidence for pockets of unburned material behind the detonation front (Courtesy ANL).