3. Research Program Accomplishments

The central goal of the Center is the detailed, whole-system simulation of solid propellant rockets under both normal and abnormal operating conditions. Full simulations of such complexity require a sequence of incremental developments—in engineering science, computer science, and systems integration—over an extended period of time. From the outset, however, our emphasis has been on system integration rather than separate threads of development that eventually come together at some point in the future. Rapid exploration of critical system integration issues entails the use of simplified—but fully integrated—models and interfaces initially, followed by successively refined models and interfaces as experience is gained.

3.1 Simulation Roadmap

The project roadmap depicts the evolution of increasingly sophisticated computational models for the primary rocket components and their interactions. The “Gantt Chart” that accompanies the Roadmap (Fig. 3.1) indicates the time sequences required for the execution of the technical program. We have been remarkably successful in completing the tasks outlined for Year 1. Completed tasks are noted on the Gantt Chart (Fig. 3.2).

The first implementation of an integrated simulation code (GEN1), fully operational in Years 2-3, provides a simplified characterization of various burn scenarios to the onset of component failures. The GEN1 code employs macroscopic models for the separate components to enable a strong, early focus on the definition and resolution of system integration issues. Refined, multiscale component models and advanced system integration concepts, based on lessons learned from GEN1, constitute the key features in the second generation (GEN2) code—targeted for Years 4-5. The refined models also reflect the synthesis of fundamental, subscale studies (bottom right side), which are critical for detailed simulations of accident scenarios and for reliable simulation of multiscale phenomena such as combustion and turbulence. The code numbers in the diagram indicates dependence of the refined and accident models on the subscale simulations.

The roadmap schematic also indicates the elaborate interconnections among the components; physical quantities such as temperature (T), mass flow (m), pressure (p), heat flux (q), concentrations (c), and geometry must be exchanged between the SRB component models. The computer science integration efforts define the framework for these interconnections and, consequently, their eventual impact on overall code performance. In the right-center box on the diagram, computer science research and development activities are shown that support the SRB simulation through the implementation and optimization of the component models and subscale simulations, the integration of component models and the computational infrastructure required to do large scale parallel computation.

Finally, the central placement of validation efforts on the diagram highlights the priority assigned to this activity. Each subscale, component, and integrated simulation must be validated against existing analytical, numerical, and experimental data available in the open literature or obtained from our sister Center for Novel Energetic Materials (CNEM).
### Research Program Accomplishments

<table>
<thead>
<tr>
<th>Structures and Materials</th>
<th>Fluid Dynamics</th>
<th>Combustion and Energetic Materials</th>
<th>Computational Math and Geometry</th>
<th>Computational Environment</th>
<th>System Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Analysis</td>
<td>Core Flow</td>
<td>Device Specification</td>
<td>Linear Solvers</td>
<td>Parallel Programming Environment</td>
<td>GEN0, GEN1, GEN2</td>
</tr>
<tr>
<td>Obtain codes</td>
<td>Moving structured mesh</td>
<td>Specify target system/components</td>
<td>Develop object-oriented framework</td>
<td>Baseline framework</td>
<td>GEN0</td>
</tr>
<tr>
<td>Implement structural elements</td>
<td>LES models in GEN1</td>
<td>Compare SDTA simulations</td>
<td>Preconditioners</td>
<td>Multiparadigm progr. environment</td>
<td>GEN1</td>
</tr>
<tr>
<td>Failure initiation models</td>
<td>Complex geometry support</td>
<td>PCI Modeling</td>
<td>Mesh Generation and Refinement</td>
<td>Code optimization</td>
<td>GEN2</td>
</tr>
<tr>
<td>Code optimization</td>
<td>Rad. and Lagrangian particles</td>
<td>Moving surface PCI</td>
<td>Arbitrary tetrahedral meshes</td>
<td>Distributed computing technology</td>
<td></td>
</tr>
<tr>
<td>Solid-Combustion-Fluid Interface</td>
<td>LES Model Evaluation</td>
<td>Preliminary 2- and 3-D codes</td>
<td>2D &amp; 3D Delaunay meshes</td>
<td>Parallel I/O</td>
<td></td>
</tr>
<tr>
<td>Fluid/structure interaction</td>
<td>Insert PCI in GEN1</td>
<td>Geometric interface model</td>
<td>Silver exodus algorithm</td>
<td>Adaptive I/O policies and software</td>
<td></td>
</tr>
<tr>
<td>Mesh adaptivity</td>
<td>LENS particles</td>
<td>Modeling of DDFT</td>
<td>Time-space meshing strategies</td>
<td>Parallel Performance Analysis</td>
<td></td>
</tr>
<tr>
<td>Initial SCFI codes</td>
<td>LENS Model Evaluation</td>
<td>Unit cell models</td>
<td>Neighboring grid interfaces</td>
<td>Next-generation tools</td>
<td></td>
</tr>
<tr>
<td>Failure Analysis</td>
<td>Evaluate LENS implementations</td>
<td>Modelling ensembles of cells</td>
<td>Anisotropic meshes</td>
<td>Train CSAR staff on architecture</td>
<td></td>
</tr>
<tr>
<td>Multiscale strategies/algorithms</td>
<td>GEN1 LES validation</td>
<td>Rocket Motor Instability Analysis</td>
<td>Adaptive refinement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial algorithm development</td>
<td>Turbulence Data Acquisition</td>
<td>Analyze thermo-acoustic instab.</td>
<td>Interpolation at Interfaces</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subscale analysis codes</td>
<td>DNS of model flows; PIV exp.</td>
<td>Provide test problems</td>
<td>Data transfer across boundaries</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Completed task (30 September 1998)
GEN0 Rocket Simulation Completed

A GEN0 exercise was performed to enable our System Integration Team to gain experience quickly with the central issues to be faced in designing and implementing the GEN1 interface code. The choice of a suitable problem was complex—it needed to be realistic enough to be instructive, but simple enough to do reasonably quickly. After careful consideration, we chose to base GEN0 on a “2-D Ideal Rocket.” The solid propellant is assumed to be homogeneous and uniform. It is described by a finite element elasticity model, loaded by the chamber pressure. The compressible fluid flow in the chamber is described by the Euler equations. The boundary conditions represent mass injection from the combustion interface. The combustion interface between solid and fluid regresses at a known rate dependent on the chamber pressure.

This problem was simple enough to be done in short order—initiated in January, completed in May—yet provided valuable experience in integrating the different component codes. It forced us to begin investigating issues such as dealing with the different time resolutions appropriate for advancing each component. The physical parameters used were derived from the Space Shuttle solid rocket booster. In addition to shedding light on various issues in component integration, the resulting GEN0 code provided the Computer Science Group with an initial starting point for performance optimization and other studies.

A small, focused team was assembled from among the Science Steering Committee, the first research scientists and programmers hired by the Center, and a few graduate students. The GEN0 team worked closely to identify the key elements of the propellant, core flow, and combustion interface required for the simple model. At least as important as the model results achieved were three lessons learned in this exercise: a small team of developers working closely together is an effective organizational strategy; mixing programming languages can be troublesome (but can be overcome); and visualization is essential for ensuring that both the problem formulation and implementation are correct.
3.2 Structures and Materials

Group Leaders: Robert Dodds, Philippe Geubelle, and Keith Hjelmstad

Faculty: Robert Averback, Armand Beaudoin, Robert Dodds, Philippe Geubelle, Robert Haber, Harry Hilton, Keith Hjelmstad, Dennis Parsons, Kenneth Schweizer, Petros Sofronis, and Daniel Tortorelli

Research Scientists/Programmers: Amit Acharya, Michael Breitenfeld, Guoyu Lin, and Nahil Sobh

Post-Doctoral Associate: Karsten Albe

Graduate Research Assistants: Dinesh Balagangadhar, Sharon Grelecki, Jason Hales, Schalk Kok, Dhirendra Kubair, Scott Meyer, Alireza Namazifard, Jayandran Palaniappan, Christian Schranz, Ertugrul Taciroglu, and Lin Yin

Visitor: Nikolaos Aravas (University of Thessaly, Greece)

Overview

The Structures and Materials Group addresses three key areas. Structural Analysis selects from existing large-scale codes, modeling software, and post-processing software to establish the best starting points to develop centerwide capabilities for full-scale stress analysis for the whole rocket. Solid-Combustion-Fluid Interface develops the strategy, theory, and key numerical algorithms needed to interface the finite element code developed through Structural Analysis with the combustion and fluid mechanics parts of the whole-rocket simulations. Failure Analysis initiates development and implementation of modeling schemes to perform detailed investigation of failure processes in the various rocket materials, e.g., metals, composites (fiber reinforced polymeric and metal matrix composites) and solid propellant.

Structural Analysis

The group led by Parsons and Hjelmstad, aided by Hales, Namazifard, Taciroglu and Schranz, has developed a scaleable parallel finite element code capable of forming the backbone for structural model development of the solid rocket motor. The code is based on a linear multigrid solver, and is capable of solving nonlinear transient problems using an implicit time integrator. The code executes in parallel on shared or distributed memory machines using the standard MPI library. Preliminary benchmarking studies demonstrate that the procedure is scaleable. These data were fit to the standard Amdahl’s law curve also shown in the figure, indicating that about 98% of the code executes in parallel. This figure is expected to increase when larger problems are considered (partly because the communication time will represent a smaller fraction of the total time).

Additional work of this group has focused on development of mesh generation procedures for the multigrid solver. We currently use TrueGrid to produce the sequence of meshes required by the multigrid solver. This allows us to generate nested meshes for complicated parts. Domain decomposition is then performed on the coarsest mesh using Metis. This ensures that we have perfect load balance among all processors on all levels. Data are finally written in the form required by our analysis code. Figure 3.4 shows some representative meshes and partitions for the rocket simulation. Other work has considered advanced solid
element formulations (e.g., based on assumed strain methods) and material model development (linear and nonlinear viscoelastic models).

In the immediate future, the Structural Analysis group will continue benchmarking studies to determine whether the code scales well on the multiprocessor ASCI computers when extremely large problems (10 million or more elements) are considered. Element and material model development will also continue. Meshing algorithms will be modified to treat the complications introduced by moving boundaries (specifically, a combustion front moving through the propellant). Contact algorithms will be combined with our multigrid solver so that details such as joints may be incorporated into the model. Studies on fluid-structure interaction will be conducted to help determine the most suitable algorithm for coupling fluid and structural calculations into a workable code.

**Solid-Combustion-Fluid Interface**

The group led by Haber and Tortorelli, with Acharya, Sobh, Lin and Palaniappan has been working this year on the formulation and implementation of advanced numerical methods for rigorous modeling of the response of the solid grain as it interacts with the moving combustion front. Part of this effort involved the formulation on nonlinear viscoelastic constitutive models for the bulk response of the solid propellant. This group has also actively participated in the on-going effort to use rigorous balance laws and constitutive relations to formulate the jump conditions at a sharp combustion interface, including the effects of finite deformation of the solid. The requirement to maintain independent descriptions of the velocities of the combustion front and the adjacent solid material is a significant outcome that distinguished the numerical solid model we are developing from the models used in earlier versions (GEN0 and GENH) of the CSAR code. This group has worked specifically on the development of space-time finite element methods that can continuously track the motion of the combustion interface. This work includes investigation of novel discontinuous Galerkin finite element formulations and 4-D space-time mesh generation with colleagues in Computer Science (Edelsbrunner and Teng). This group is exploring variants of Finite Element Tearing and Interconnecting method (FETI) algorithms as a means of enforcing the jump conditions across the combustion interface in the coupled system, as well as domain decomposition techniques for parallel computation within the solid region.

Fig 3.4  Successively refined meshes for Space Shuttle SRB.
In the coming year, the Solid-Combustion-Fluid Interface group will develop new codes based on these formulations to describe the response of the solid propellant during a normal burn scenario. The resulting code will be integrated with codes for the response of the fluid, combustion and case regions. Scalability will play a central role in the software design. Work will also be done on adaptive procedures to take advantage of unstructured meshing over the space-time domain.

**Failure Analysis**

In addition to research activities directly concerned with the development of the fully integrated (GEN1) code, a series of research efforts related to the GEN2 code are currently underway. Most of these activities are aimed at modeling the constitutive and failure behaviors of the solid propellant and the case, and involve a wide variety of numerical tools.

At the continuum level, Sofronis and Meyer are developing a 3-D phenomenological model for the behavior of the polymeric binder in the viscoelastic regime. The model, which accounts for large strains and for the effect of strain rate and temperature, has recently been implemented into an ABAQUS UMAT subroutine and is currently being tested under various loading conditions. Future efforts include combining this model with a special form of Hill’s approach aimed at capturing the failure behavior of the solid propellant. After the constitutive law for nonlinear response is established, the issue of hot spot generation will be addressed. The idea is that intense shear deformation within the material may result in rapid energy generation that can trigger a localized reaction leading to a hot spot. These phenomena of plastic instability in solid propellants are investigated in collaboration with Professor Aravas of the University of Thessaly, Greece, who visited CSAR this past summer.

Also at the continuum level, Geubelle and members of his group are developing a special form of the cohesive volumetric finite element (CVFE) scheme to study spontaneous crack propagation in the solid propellant. Special emphasis is put on the incorporation of rate dependence in the failure model and in the adequate treatment of contact between the crack faces. Kubair currently investigates the importance of rate-dependence in spontaneous dynamic cohesive fracture with the aid of a spectral scheme. Lin has developed a quasi-static CVFE scheme, which includes a local contact component. To validate the code, numerical and experimental results a fiber pushout test performed on a model polyester/epoxy composite system have been compared. Excellent agreement has been shown, as illustrated in Fig.

![Fig. 3.5: Punch force vs. displacement curve (left) and evolution of the debond length (right) for a quasi-static fiber pushout test performed on a polyester/epoxy model composite system. The symbols correspond to the experimental observations, and the solid curve, to the numerical simulations recently performed by Lin and Geubelle with a special form of the cohesive/volumetric finite element scheme.](image-url)
3.5. Future plans for this part of the project include the addition of a global contact algorithm in both the quasi-static and dynamic CVFE codes.

Beaudoin, Dodds, Acharya, and Schalk are currently working on the formulation and implementation of a physically motivated gradient single crystal plasticity model to explain size effects in metallic single crystal response. Special emphasis has been put so far on the implementation of the Mechanical Stress Threshold (MTS) model. At present, a material subroutine has been written for rate-dependent, isothermal yield behavior, incorporated into ABAQUS and used to simulate stress relaxation for aluminum alloys (Fig. 3.6). A literature search for the D6AC steel used in the Space Shuttle has provided a report on biaxial testing, authored by the Defense Science and Technology Organization of Australia. The experimental data available in that report will be used to fit the parameters to the MTS model for D6AC. Future plans involve extending the formulation to polycrystalline plasticity and incorporating the final model in existing 3-D ductile fracture codes such as WARP3D. This part of the project also involves close collaboration with Dr. Stout’s group at LANL where Beaudoin and one of his students (Michael Bange) spent a substantial part of this past summer collecting high strain rate and high temperature data on aluminum alloys.

At a much smaller scale, Averback, Sofronis, and Albe have recently started to develop a coupled continuum/molecular dynamics scheme to study the debonding of polymeric-ceramic interfaces. As a first step, interfacial fracture between metals and their native oxides will be studied as model for interfacial fracture in a rigid/plastic system. Work is currently underway to develop the appropriate potentials for this system.

Finally, molecular dynamics (MD) simulations are also used by Schweizer to study the tracer diffusion of penetrants in dense polymeric materials. This phenomenon is of fundamental importance in a variety of material reliability and aging problems. It is a slow process, and thus difficult or impossible to simulate at the atomistic level via classical MD techniques. A project has been initiated to develop a novel hybrid theory-simulation approach that combines short time MD results with nonequilibrium statistical dynamical theory for long time motion. The goal is to predict for specific material systems both the macroscopic penetrant diffusion constant and anomalous non-Fickian transport at intermediate times. In collaboration with scientists at Sandia National Laboratory, future work will continue to focus on methodology development, and also quantitative tests of the hybrid approach by comparing its predictions against massive simulations for model systems consisting of small spherical penetrant particles diffusing in polymer melts of variable local structures.

3.3 Fluid Dynamics

Group Leaders: Robert Moser and Michael Norman
Overview

The Fluid Dynamics group works on system-scale solid rocket motor core flow model development as well as subscale model development relevant to the turbulent dynamics of the combustion interface, dispersion and combustion of Al particulates in the core flow, and crack flow. A major focus is the development of appropriate large eddy simulation (LES) models for the core flow using insights gained from direct numerical simulation (DNS) and particle imaging velocimetry (PIV) laboratory experiments.

Core Flow

Balachandar, in collaboration with Short and Buckmaster, has analyzed non-axisymmetric fluid flow in solid rocket chambers. They have shown that modest deviations from symmetry, arising from cross-section perturbations, variations in burning rate, etc., lead to the generation of a strong axial vorticity field in the neighborhood of the axis. The magnitude of this vorticity is an increasing function of the flow Reynolds number. These results may have serious implications for the accurate calculation of the turbulent flow in the chamber when idealized symmetry is not achieved.

Aref and Pushkin explored analytical solutions representing vortex motion in compressible flows. There were at least two motivations for this. First, we were interested in the problem of the fluid exiting the rocket chamber. Since the fluid is turbulent, it will contain many vortex structures. However, in most work on rockets the effect of such vortex structures is ignored in the nozzle exit flow. A second motivation was the comparison with numerical simulations of compressible flow. It turns out that vortex structures in compressible flow have complex structure, such as shock waves, within them. In a numerical simulation this structure presumably needs to be resolved or at least modeled in an appropriate fashion. We hoped that by identifying analytical solutions we could use these for testing of computer codes and for order of magnitude assessment of various resolution issues that arise in numerical simulations. After a literature survey, Pushkin was able to derive a general formalism that captures the known cases studied in the literature. However, we were not able to find simple analytical forms for some of these solutions (as can be done in the incompressible case).
Alavilli and Tafti are developing *ROCFLO*, a new code for 2-D/3-D SRM core flow simulation and a component of the GEN1 system code. *ROCFLO* is a multi-block, compressible, inviscid (Euler) flow code for both steady and unsteady computations. Within a block, spatial discretization utilizes a structured, body-conforming mesh. The integral conservation laws are solved within control volumes using an explicit multistage Runge-Kutta time stepping scheme and a variety of spatial differencing schemes. Moving grid features are incorporated into the governing equations. These include flux corrections due to grid movement and source terms due to cell volume changes. A serial (F90), time-explicit, multi-block implementation of *ROCFLO* has been completed and work is underway on a parallel version (F90/MPI). A test computation of the Shuttle SRB is shown in Fig. 3.7.

Tafti has explored single processor and parallel optimization strategies for CFD codes and relevant linear system solvers on microprocessor based architectures including the SGI Origin2000. These findings will be used to optimize *ROCFLO* for ASCI systems.

The multiblock capabilities of *ROCFLO* will support the complex, regular geometries of an SRM (i.e., star grains, inhibitors, etc.). As burn proceeds, blocks will need to be redefined to maintain geometric regularity as well as computational efficiency. *ROCFLO* was implemented in F90 because of its support for dynamic creation, deletion and merging of computational blocks as the simulation progresses. F90 features are also invaluable in developing local adaptive mesh refinement strategies.

Liou and Balsara are developing an unstructured adaptive mesh fluid dynamics code for eventual integration into GEN2. The principal application of the unstructured mesh code will be fluid flow in cracks as they develop in failure scenarios. Additionally, an unstructured code may prove useful for meshing geometrically difficult portions of the flow or at receding boundaries. A 2-D code has been developed as a testbed to explore certain algorithmic features desirable in the final code, including monotonicity preservation at shock fronts, a multifluid Riemann solver for reactive flow, moving boundaries, and multigrid acceleration for convergence to steady state.

A variety of research projects, both analytical and numerical, have been initiated to better understand the micro- and macroscopic fluid dynamics of the multiphase core flow that is produced by the combustion of aluminized solid propellants. Many of these projects are done in collaboration with members of the Combustion and Energetic Materials Team. Findings from these projects will provide the physical basis for constructing the GEN2 core flow model.

Bagchi, Balachandar, and Ha, in collaboration with Stewart, Krier, and Brewster are constructing a subscale simulation of aluminum droplet combustion inside the rocket chamber. The objective is to obtain a comprehensive understanding of the combustion process, and develop models for burn rate, drag, etc., as a function of particle Reynolds number and surrounding flow environment. These models will be used in the GEN2 whole system simulation to account accurately for the volumetrically distributed exchange of mass, momentum and energy between the Al droplets and the surrounding core flow. Detailed 3-D microsimulations of the combustion of aluminum droplets will be performed. Effects of droplet Reynolds number and critical parameters describing the surrounding medium such as temperature, species concentration and local flow gradient will be investigated. The Reynolds number of the flow inside the droplet is expected to be larger than that outside, as a result the
coupled problem of flow both inside and outside the droplet will be considered. The effect of radiative heat transfer on droplet temperature distribution and hence combustion rate will also be eventually included in the analysis. In the past six months a comprehensive 3-D code has been under development. The code development, testing and parallelization efforts will require another six to nine months. Subsequent production runs will develop appropriate Al particulate combustion models.

Ferry and Balachandar are studying particle dispersion in the core flow. This project has two objectives. First, to determine the feasibility of an Eulerian (i.e., continuum) representation of the particles, which may be more efficient than the straightforward, Lagrangian approach. Second, to apply the optimal LES formalism, using it to test candidate subgrid-scale models. This involves calculating the requisite two-point correlations and also quantifying how the particle distributions differ when they are evolved using filtered velocity fields of varying coarseness. Work so far has consisted of porting a channel flow code to a parallel environment. Parallel optimization is currently underway.

Vanka and Mukhopadhyay have developed a 3-D incompressible CFD code that will be used to simulate turbulent pipe flow with particles. The code employs a finite volume formulation of the Navier-Stokes equation with staggered locations for velocity components and pressure. Temporal integration uses an Adams-Bashforth scheme. The elliptic pressure equation is solved using FFTs in the axial and angular coordinates and line inversion in the radial direction. Various LES prescriptions from the literature as well as those developed by R. Moser’s group will be employed. The project has two thrusts: in one effort, LES simulations of flow in a pipe with realistic L/D will be carried out, including transverse injection of fluid. Results have been obtained for a series of simulations with injection Reynolds numbers of 10, 20, 40, 45.55, 80, 120 and 160. These simulations show that at these low Reynolds numbers the 3-D flow remains axisymmetric. In the second effort, DNS and LES simulations of particle-laden flows will be carried out to study the effect of particles on the turbulence.

Under separate ASCI funding, Norman and Hayes are exploring the performance and parallel scalability of numerical algorithms for 3-D radiation hydrodynamics simulations on ASCI Tiflop architectures. This effort will provide the basis for the radiation transport solver needed for the GEN2 core flow code. The ZEUS-MP code has been developed as a testbed implementing both explicit and implicit algorithms to solve the radiation diffusion equation, which is then coupled to the Euler equations of hydrodynamics. The efficacy of multigrid and preconditioned conjugate gradient methods are compared for solving the system of linear equations resulting from the implicit scheme. Preliminary time-to-solution studies indicate the implicit algorithm may be preferable to the explicit algorithm for optically thin, strongly coupled media. However this may change as problem sizes are increased. Near ideal scaling of both algorithms is found to 128 processors on a SGI/CRAY T3E supercomputer—the largest configuration attempted to date. On the SGI/CRAY Origin2000, speedup is sublinear indicating network bottlenecks. The single node performance of the Origin2000 processor is roughly 2x the speed of the T3E. Scaling studies to >1000 processors of the ASCI Red machine are currently being performed.

**LES Model Evaluation**

Moser, Bagchi and others have been pursuing the development of LES models applicable to the solid rocket core and nozzle flow. A new technique for developing formulations is being
applied, in which the LES formulation is determined through a formal optimization. The result is a LES model that best represents large-scale turbulence dynamics and assures that the large-scale statistics match those of real turbulence. The technique has been applied to isotropic turbulence with the surprising result that only the scale-dependent dissipation can be modeled correctly, and that this is enough to generate an accurate LES. This provides a theoretical explanation for previously observed properties of LES models.

Application of these techniques in strongly inhomogeneous flows requires development of appropriate inhomogeneous formulations. This is being pursued using data for a turbulent channel flow. By careful application of inhomogeneous filters and using the optimization procedure discussed above, it was found that the filter inhomogeneities caused the magnitude of the model term to be orders of magnitude larger than in the isotropic case. It was also found that this large model term was essentially unpredictable, making a LES with such filter a questionable undertaking. These results suggest the importance of filter definitions in these highly inhomogeneous flows. For the later the DNS data being obtained by Najjar will be particularly valuable.

Najjar and Balachandar are also working on the development of optimal LES models for complex flows. The optimal LES formalism has so far been developed and tested in only simple flows, such as isotropic turbulence and turbulent channel flow. The application of optimal LES formalism to the rocket flow is likely to be far more complex, owing to directional inhomogeneity, geometric complexity, strong wall transpiration, two-phase nature of the flow, etc. The development of near optimal LES models requires information on unconditional statistics in the form of two-point correlations. Lack of homogeneous directions in the actual rocket flow will place severe restrictions on the size of two-point correlation and on the number of data samples available for model development and testing. New approaches (different from those taken in the simpler problems) are required to cope up with these restrictions. Here we have developed a new approach for the application of optimal LES formalism to complex problems with multiple directions of inhomogeneity. This approach relies upon projection and filtering in the eigenspace along the inhomogeneous directions. It has been tested in a problem with two inhomogeneous directions. The efficacies of optimal linear and quadratic eddy viscosity and Smagorinsky type models have been tested.

**Turbulence Data Acquisition**

Adrian and Tomkins are concentrating on two areas relevant to CSAR: the design of laboratory experiments to investigate turbulent flows within a solid rocket engine, and the analysis of structures in wall turbulence. The experiments will produce data over a range of Reynolds numbers; this will allow for comparison with direct and large eddy simulations, while providing results at higher speeds closer to actual operating conditions. Experiments are planned in both channel and pipe configurations. For the former we have designed modifications of an
existing wind tunnel to create a 2-D channel flow with one end sealed and blowing through one rough wall. For the latter we have designed and are ordering a pipe flow apparatus with blowing through a porous section, as shown in Fig. 3.8. It allows for two operating scenarios: purely injection-driven flow in the test section of the rocket core with the upstream inlet sealed, as in the actual engine, and fully developed turbulent pipe flow entering the test section where it encounters wall injection.

### 3.4 Combustion and Energetic Materials

**Group Leaders:** M. Quinn Brewster, Herman Krier, and D. Scott Stewart

**Faculty:** Quinn Brewster, John Buckmaster, David Ceperley, Eliot Fried, Herman Krier, Richard Martin, Todd Martinez, Mark Short, and Scott Stewart

**Research Scientists/Programmers:** Thomas Jackson, Min-Gyoo Lee, Lubos Mitas, Kung-Chyun Tang, and Shaojie Xu

**Post-Doctoral Associate:** Uwe Stephan

**Graduate Research Assistants:** Gregory Knott, Serguei Kochevets, Vladimir Korchagin, Igor Kouznetsov, Paul Loner, Kanishk Mahajan, John Melcher, Burkhard Militzer, Jeffrey Murphy, Lisa Orth, Tommaso Torelli, Jack Yoh, Fenghua Zong

**Visitor:** Jose Soler (Universidad Autonoma de Madrid)

**Overview**

Combustion of solid propellant is the driving mechanism in the operation of solid rockets. Correct description of the solid propellant regression rate is critical to the success of the multi-disciplinary, whole-system simulation of solid rocket motors. Subsequent simulations of fluid dynamics, turbulence, heat transfer, structural dynamics, and the interactions among different mechanisms depend on regression rate. Accurate knowledge of the chemistry associated with the combustion rate (or burning rate) of the solid propellants is important. Chemical processes determine these rates, both condensed phase and gas phase. Also necessary are descriptions of the transport of heat, mass and species across the propellant-combustion interface. In addition, fundamental atomic and molecular physics of the transition states of energetic materials must be an integral part of the simulation. Preliminary efforts will be made in research and code development of aluminum droplet combustion, gas and particle radiative heat transfer, and aluminum distributed two phase reacting flow analyses.

**Device Specification**

In collaboration with NASA and our industrial colleagues, CSAR has selected the solid rocket boosters (SRB) of the Space Transportation System (STS)—better known as the Space Shuttle—as the device for our initial simulation. The Shuttle SRB is a well-established, commercial rocket, is globally recognized, and most importantly, basic design data and propellant configurations are available through NASA and Thiokol.
**Modeling of Propellant Combustion Interface**

The propellant combustion interface (PCI) is the layer of combustion that separates the solid propellant from the core gases. The PCI moves normal to this interface at the nominal regression rate for the propellant. Stewart, Jackson, Xu, Fried, and Short have a project underway that represents the motion of the PCI using the method of level sets. The rules for self-consistent motion of the surface are being constrained with the use of continuum mechanical analysis of generalized jump conditions for thermomechanical materials. The solid, sufficiently far from the PCI, is assumed to be a generalized, heat-conducting solid, and the fluid gases in the core, sufficiently far from the PCI, is assumed to be an inviscid, compressible gas. The normal jump conditions consider the heat release in the PCI and show the expected jump conditions for conservation of mass, momentum and energy, plus an additional constraint on the functional form of the regression rate argued from the dissipation inequality and the surface kinematic relations. The level-set surface representation plus the thermodynamically self-consistent conditions form the PCI model. Related studies are underway to check that conventional combustion models that included detailed models of sub-scale physics, such as pyrolysis and grain-level flame physics, have far field averaged solutions that are consistent with the macroscale constraints on the regression rate.

A computer code for the arbitrary surface topologies that can interpolate near surface fields from different meshes on either side of the PCI is being designed and written that will be capable of representing the PCI between the solid and fluid system code components. Early collaborations on the level-set methods this past year have included exploratory work with Tariq Aslam at Los Alamos National Laboratory and prototyping 3-D level set code after Aslam’s code, *DKAPPA3D*. Figure 3.9 shows a very early burnout of a propellant grain, computed with *DKAPPA3D*, which regresses the PCI at a normal speed near 3 cm/s.

![Fig. 3.9. Initial burnout simulation for SRB grain using DKAPPA3D.](image)
Short, in collaboration with Dr. James J. Quirk, a member of the Caltech ASCI team, has been examining effective ways of solving chemical flow problems with adaptive mesh refinement and parallelization. This has implications for the full-scale rocket simulation, in which the use of adaptive mesh refinement could substantially lower the cost of computing the model chemistry and complex flow geometries involved.

**Rocket Motor Instability Analysis**

Brewster, Loner, Tang, O’Shea, and Knott worked closely in several instability analysis projects. These included developing a nonlinear, unsteady propellant (homogeneous, nonmetalized) combustion model; establishing a coupled propellant combustion and simplified motor chamber model for \( L^* \) instability and generalized (i.e., homogeneous and composite) nonlinear, unsteady propellant combustion; designing a \( L^* \) burner to perform experimental work to obtain modeling data necessary for simulating unsteady composite propellant combustion; and beginning the analytical modeling of composite propellant combustion.

The group’s most significant single research finding is preliminary numerical prediction of oscillatory nonlinear \( L^* \) behavior—which has been observed experimentally, but never simulated computationally—involving initial high-frequency oscillations early in the firing at low pressure followed by low-frequency oscillations at high pressure. Other significant results include numerical simulation of complex extinction and chuffing behavior.

\( L^* \) instability is the phenomena of low frequency oscillatory combustion coupled with spatially uniform chamber pressure fluctuation occurred in solid rocket motors with small characteristic length \( L^* \) (defined as the ratio of free chamber volume to nozzle throat area). Numerical simulation of \( L^* \) instability for homogeneous propellant in solid rocket motors has been performed. The model of a simplified kinetics combustion model coupled with a simplified \( L^* \) combustor is used. The combustion model, which was developed by Ward, Son, and Brewster (WSB), of a new low gas activation energy analysis in the gas phase with the activation energy asymptotic thermal decomposition analysis of Lengelle in the condensed phase is used for modeling the combustion of homogeneous propellants. The quasi-steady gas and condensed phase reaction (surface reaction) were assumed in the modeling of un-
steady nonlinear combustion and gas dynamics. Several nonlinear behaviors are predicted computationally for the first time, many of which are similar to observed nonlinear L* instability behavior such as extinction (Figure 3.10) and chuffing (Figure 3.11\(^1\)). Frequency shifting has been observed experimentally in double base propellant but attributed tentatively to the two-stage flame structure of these propellants or incomplete combustion due to short gas resident time. In the present model, the initial high-frequency behavior is manifested without including a two-stage flame structure or the modeling of incomplete combustion. It therefore appears that this behavior may be due to other effects, which are currently under investigation.

Krier and Lee are performing numerical simulation of combustion instability in solid-propellant rocket motors. Combustion instability associated with the pressure dependence of the burning rate of energetic materials in solid-propellant rocket motors can lead to catastrophic failure. Combustion instability in a solid rocket motor chamber are mainly due to the interaction between the unsteady energy released by exothermic chemical reactions of energetic materials and the combustion chamber wave dynamics. Typical combustion instability can often be characterized by oscillatory pressure fluctuations coupled with unsteady heat release. The unsteady combustion of solid-propellant (SP) is highly responsive to the 3-D pressure fluctuation of high-temperature burnt gas in a rocket motor chamber. The contributions of the heat release due to SP combustion to the acoustic energy in the chamber can be described by an acoustic admittance function. For a quasi-steady gas flame, the acoustic admittance function is directly related to the propellant burning rate response function. For a rigorous description of combustion/acoustic instability in rocket motor chambers, the physical aspects such as combustion response of SP, compressibility of burnt gas, chamber volume changes due to burn-back of SP grain, and the large shear and temperature stresses along the burning interface have to be considered.

The objective of the present study is to analyze the combustion instability for both homogeneous and heterogeneous solid propelants in large motors. Murphy and Krier have extended recent unsteady burning models to heterogeneous propellants. Their statistical response function for heterogeneous SP will be implemented in the present numerical simulation. The fluid flow with nonstationary waves, governed by unsteady compressible Navier-Stokes equations, has been numerically solved by the third-order approximate Riemann solver, Flux Vector Splitting, and Advection Upstream Splitting Method, with MUSCL (Monotonic Upstream centered Scheme for Conservation Laws) extrapolation. To take account of moving SP grain, the so-called Geometric Conservation Law for a moving grid and the two-block structured grid system have been employed.

To investigate combustion instability in a solid-propellant rocket motor, a high-order Navier-Stokes solver (third-order in space and fourth-order in time) with a moving boundary capability has been developed, which is the first stage of combustion instability analysis code development. The preliminary numerical calculation for unsteady chamber filling process has reported. In those calculations a phenomenological burning rate model, i.e., \( r_b = a p^n \), and non-moving interface model were adopted. The quasi-steady numerical solutions have been compared with a simple gas dynamics theory, and satisfactory agreement was achieved. The

---

\(^1\) Chuffing is the phenomenon of re-ignition following extinction for many times. Although extinction of combustion is predicted by the simulation, the code cannot simulate re-ignition following extinction.
steady-state flowfield solution has also been perturbed by introducing a sinusoidal forcing function at head-end wall to investigate the acoustic wave dynamics inside a rocket motor chamber. In the ongoing stage of the numerical code development process, the combustion response function of heterogeneous SP is currently being implemented and parallelism introduced into the code.

**Modeling of Solid Propellant Flames**

Buckmaster and Jackson have collaborated in research on idealized propellant flame configurations for the purpose of identifying the key modeling ingredients needed for accurate simulations. The most significant result is the demonstration that a time dependent shear in the neighborhood of the propellant surface of the kind generated by the interaction of acoustic waves and the mean flow can generate large variations in the heat flux to the propellant, both instantaneous and time-averaged, with concomitant effect on the regression rate. There appears to be no literature on non-axisymmetric vortical rocket chamber flows, and so there are a large number of questions to be examined. They plan to increase the complexity of the propellant flame modeling, accounting for 3-D effects, more complex kinetic models, etc.

![Fig. 3.12. Large variation in heat flux caused by the interaction of acoustic waves and mean core flow.](image)

**Constitutive Modeling of Energetic Materials**

Fried has focused his attention on the formulation of a thermodynamically consistent description of the propellant combustion process, a description in which the interface between solid rocket propellant and its combusted product is modeled as a sharp surface of discontinuity in various relevant bulk fields and constitutive properties. The key aspect of this work concerns the formulation of appropriate kinematic, kinetic, and constitutive equations for the interface. This effort has been conducted using modern continuum mechanical methods developed for the study of phase transitions. The investigator has interacted closely with other members of the combustion team, including Jackson, Stewart, and Xu. Thus far, this work has yielded a 1-D model of a rocket consisting of a system of governing equations that involve bulk and interfacial partial differential equations.

During the coming year, Fried will work with Jackson and Stewart to analyze the aforementioned system of equations with a view to comparing their solutions with those obtained from standard diffuse-interface models for combustion. The goal is to obtain guidance with regard to the structure of interfacial constitutive equations that describe the kinetics of the combustion process and the evolving jump in the temperature field across the combustion front in the sharp-interface model. In addition, they will continue to work on a full 3-D sharp-
interface description of the combustion process, with a focus on including in the theory the
effects of interfacial energy, stress, and other thermodynamic fields. With Vanka, a turbu-
rence model is to be included. Solvers for structural mechanics and heat conduction in SP
grain will also be implemented into the code in the near future and then integrated into GEN1
or GEN2.

Fermionic path integral Monte Carlo (FPIMC) simulations have been used to study
the equilibrium properties of the hydrogen and deuterium in the density and temperature
range of \( 1.6 < r_s < 14.0 \) and \( 5000K < T < 167000K \). Ceperley and Militzer studied the nature
of the transitions and used a cluster analysis to identify the various chemical species in each
phase such as H, H+, e, H2 and H2+. They also determined the equation of state of deuterium
along the Hugoniot, which is studied by laser shock wave experiments by the Lawrence
Livermore group. A recent news article in *Science* (218, 1135, 1178, Aug. 21, 1998) has a
description of the experiments and compares to their calculations. Ceperley attended a work-
shop at Livermore and was one of the presenters on the properties of high pressure-high den-
sity hydrogen in January 1998. The investigators have used the ACSI Blue SP2 at LLNL for
developing and running the parallel FPIMC simulations. They were one of the heaviest users
of machine this year. In particular, full parallelization and testing of the path integral code
was an important part of this year’s computational advances.

As part of his graduate studies, Militzer spent the summer at Lawrence Livermore
National Laboratory working with E. L. Pollock and others on the above research. He de-
veloped a new variational approach based on the density matrix in order to study the theory of
warm condensed matter with a particular focus on hydrogen. Next year they will develop
pseudopotentials for use with their path integral code (thesis work of Zong) so that they can
treat the heavier atoms occurring in rocket fuel and will modify the code to use the pseudop-
tentials and test them on applications.

Mitas is using Quantum Monte Carlo methods for studying electronic structure of
molecular and condensed systems with high-energy storage. Quantum Monte Carlo (QMC) is
one of the most promising methods for high accuracy calculations of atomization and cohe-
sive energies, heats of formation, excitation energies and barrier heights, etc., for both mo-
lecular and condensed matter systems. It is well known that current quantum chemistry and
condensed matter physics methods in many cases are not capable of providing the energy dif-
fferences with desired accuracy (1 kcal/mol). QMC provides a new alternative to the tradi-
tional electronic structure approaches. Some advantages:

- The QMC method relies on explicitly correlated wavefunctions and stochastic techniques
to solve the Schrodinger equation. This combination enables the description of the ex-
tremely complex many-body effects with high accuracy and efficiency.

- Because of a negligible communication overhead, QMC is ideal for massively parallel
architectures such as ASCI machines. Tests on Origin, SP2, Exemplar and T3E up to 512
processors show essentially perfect scalability.

- QMC has a wide range of applicability as demonstrated by their and others recent cal-
culations for molecules, clusters and solids. Some of these calculations belong to the largest
ever done within the correlated wavefunction framework (~ 200 valence electrons).
Mitas has also initiated application of QMC to several high-energy molecular systems, in particular, to cage molecules of carbon and nitrogen. This will involve systems such as octanitrocubane and 1,3,5,7-tetranitrocubane that are highly energetic (it is estimated that octanitrocubane can be more effective than HMX by 25% or so). Fundamental difficulties with these systems lie in the fact that either their synthesis has not been successful so far or the synthesis is too expensive. Plans for the next year are to expand the usefulness of the QMC method to calculations of interatomic forces. Such development will enable descriptions for equilibrium geometries, and open possibilities for dynamical calculations in a fully correlated manner.

Martin, Stephan, and Mattson are performing quantum simulations of hydrocarbons at high pressure and temperature. The primary goal of the project is to develop efficient methods for \textit{ab initio} simulation of materials under extreme conditions. They are developing “Order N” methods that simulate thermal motions of atoms and reactions directly from a quantum treatment of the electrons. These methods solve the quantum mechanical equations with computational complexity that scales linearly in the number of atoms. In contrast, the most efficient current methods scale as the square or the cube of the number. Order N methods are intrinsically parallel in nature and adaptable to parallel computers. They have simpler versions running that make certain approximations. The main efforts now are to improve the programs to include self-consistent density functional methods that are state-of-the-art in chemistry and physics. In this they are working with Soler and Ordejon (former member of the group). The codes are called SIESTA and presently are advanced and highly structured, but only for serial architectures. Mattson is working with these codes and Stephan is developing a more general Order-N package that can be used with SIESTA. The first applications are for the equation of state of carbon at high pressure and temperature. In the future, they will simulate reactions in hydrocarbons under extreme conditions.

Martinez leads a team that aims to achieve a first-principles description of molecular dynamics including electronic excitation that they will use to study combustion and detonation. In the first year, they have succeeded in implementing a multiple spawning method on parallel architectures, achieving near-linear speedups (up to 28 times faster on 30 processors) and have also performed the first ever \textit{ab initio} molecular dynamics simulations including electronic excitation. However, in order to be able to apply these methods to larger systems which more accurately model true materials, they need to make significant advances in the methods used to compute interatomic potential energy surfaces, for both ground and excited electronic states. They have pursued two directions here—first, combining interpolation methods with direct \textit{ab initio} evaluation of the potential energy surfaces and second, assembling intermolecular interactions from detailed information about fragments of the molecule. The first of these is most well suited to the study of large molecules, while the second is best for large systems composed of many molecules, e.g. molecular crystalline solids. They are applying the first-principles methods to investigate the reactions of ozone under high pressure and further developing their methods for solving the electronic Schrödinger equation in large systems and/or at high pressures.

3.5 Computer Science

Group Leaders: Michael Heath, David Padua, and Daniel Reed
Faculty: Herbert Edelsbrunner, Michael Heath, Laxmikant (Sanjay) Kale, David Padua, Daniel Reed, Shanghua Teng, Paul Saylor, and Marianne Winslett
Research Scientists/Programmers: Milind Bhandarkar, Jay Hoeflinger, and Kent Seamons
Post-Doctoral Associate: Mario Pantano
Graduate Research Assistants: Gheorge Almasi, Yong Cho, Cinda Heeren, Xiangmin Jiao, Szu-Wen Kuo, Jonghyun Lee, Xiangyang Li, Yuan Lin, Girish Maiya, James Oly, Ali Pinar, P. Ramachandran, Alper Ungor, Akhil Vidwans, Terry Wilmarth, and Afra Zomorodian
Visitor: Tamal Dey (Indian Institute of Technology, Kharagpur)

Overview
Research in Computer Science is addressed by two teams, Computational Mathematics and Geometry, and Computational Environment. Work in computational mathematics and geometry focuses on the areas of linear solvers, mesh generation and refinement, and interpolation between diverse discretizations. The target of our computational environment program is to provide the broad computational infrastructure necessary to support complex, large-scale simulations in general, and for rocket simulation in particular. Areas of research include parallel programming environments, compiler optimization and parallelization, parallel performance analysis and tools, and parallel input/output.

Computational Mathematics and Geometry

Linear Solvers
Work on linear solvers by Heath, Saylor, Pinar, and Vidwans has proceeded along several fronts. We have developed a new grid-based approach to linear solvers in which the grid structure is exploited directly rather than forming the corresponding sparse matrix explicitly. Such an approach avoids the overhead in both processing time and memory required for converting back and forth between the two representations, and also enables the exploitation of structural properties of the grid that are obscured in the matrix representation. To implement this new approach for direct solution, we have developed a new ordering algorithm based on Cartesian nested dissection using axial edge separators of the dual graph, edge-based algorithms for computing the elimination tree, and multifrontal algorithms for numeric factorization and triangular solution. In addition, using pair-wise disjoint subgrids, this approach naturally yields structural preconditioners that are block diagonal and easily factorized, thereby unifying direct and iterative methods within this same framework, which has been implemented in an object-oriented manner.

Other work on linear solvers has included performance enhancement of iterative methods through exploiting instruction level parallelism and improved cache behavior in sparse matrix-vector multiplication and improved error estimates for conjugate gradient methods that enable sharper stopping criteria. We have developed and studied the performance of new methods for solving sparse triangular linear systems based on selective inversion of small submatrices. These methods have exhibited excellent scalability and have important applications in direct methods as well as for triangular preconditioners for iterative methods.
Mesh Generation and Refinement

Edelsbrunner, Teng, Dey, Heeren, Li, Ungor, and Zomorodian have been working in the area of mesh generation, and especially on the construction of anisotropic meshes. Our favorite approach uses mesh refinement and simplification in an alternating fashion. We established local criteria that characterize simplification operations preserving the topology of the mesh. We expect that the implementation of these ideas will lead to fast and conveniently adaptive mesh generation code. We also have worked on the very important phenomenon of slivers in 3-D Delaunay triangulations. These are very flat tetrahedra, and experience shows that a non-negligible fraction of all tetrahedra are slivers. We have found point-weighting techniques that we have proven to be effective in eliminating all slivers. Because the sliver problem has been around for a long time, this result can be considered a breakthrough in the area.

Another important achievement in the mesh generation area is the development of a unified scheme for simultaneously refining and coarsening. Our method guarantees that the resulting mesh is well shaped and size-optimal. Finally, we have studied the problem of load balancing in parallel adaptive mesh refinement algorithms. A dynamic load-balancing algorithm has been developed to ensure proper distribution of both refinement and computation while the communication overhead is minimized. The main ingredient of our method is an efficient technique for size estimation and element distribution of the refined mesh before we actually generate the refined mesh.

A key goal for the next year will be to complete the development of the meshing software to the point where it will be able to handle the complex boundaries required for our simulation codes. We plan to develop further and implement the sliver elimination algorithm and to study experimentally the largest angle that can be guaranteed as a minimum dihedral angle in 3-D Delaunay triangulations. We also plan to implement the generation of anisotropic meshes through alternating mesh refinement and simplification. The anisotropic effect will be achieved by guiding the simplification operations using error measures modeled as Riemannian metrics. We have developed several practical variations of our provably good algorithm for adaptive meshing with a moving boundary. We will implement this class of methods and work with other CSAR researchers to define the interface between our mesh generator with the rocket simulation code. Finally, in cooperation with CSAR researchers from other areas, we will develop space-time meshing strategies.

Fig. 3.13. Interpolation between disparate meshes.
Interpolation at Interfaces

Edelsbrunner, Heath, and Jiao have developed algorithms for interpolating data between disparate grids at component interfaces. A typical example is the interface between solid and fluid in a rocket simulation. The fluid grid is typically a structured finite difference mesh with fine resolution, whereas the solid grid is typically an unstructured finite element mesh with relatively coarse resolution (Fig. 3.13). We have developed algorithms for interpolating physical quantities, such as pressure, between grids in both two and three dimensions.

Computational Environment

Parallel I/O

In the parallel I/O area, Winslett, Seamons, Cho, Kuo, and Lee have developed strategies to make parallel I/O libraries self-tuning. This enables ordinary users who are not “I/O wizards” to attain good performance even for complex applications with involved I/O patterns. Another important development deals with the fact that many applications do not spread their data evenly across processors. Unfortunately, irregular data distributions often lead to performance sapping I/O load imbalance. We have successfully developed and implemented methods to prevent I/O load imbalance, and evaluated them using data distributions from adaptive mesh refinement.

We have implemented in the Panda system the self-tuning and load-balancing strategies mentioned earlier. We also have developed an HTTP-based data migration facility for Panda that allows supercomputer users to transparently access files on remote computers. Integration of the facility with Panda allows the latency of such accesses to be hidden through appropriate caching and prefetching strategies, leading to significant performance advantages compared to manual migration. We ported Panda to an Origin2000 and a PC cluster at NCSA. The I/O performance bottlenecks that we uncovered led NCSA to reconfigure the Origin. On the cluster, we found that the use of SMPs in PCs causes resource contention that can sharply limit I/O performance, unless I/O servers are carefully placed. Finally, a pre-alpha release of Panda 3.0 that includes all the features discussed above was built. We demonstrated the package at the HPDC 98 conference using ZEUS-MP, an application obtained from CSAR application researchers.

An important research focus next year is on reducing the high cost of migrating data to and from supercomputing platforms by exploiting the Panda parallel I/O library’s self-tuning capabilities. A second focus is on compiler support for runtime parallel I/O. A compiler has much advance information about the I/O requests to be made by an application, and this information can be used by Panda to fine-tune its I/O strategies at run time, e.g., by allowing it to prefetch data before an I/O request is actually issued, thereby reducing the apparent cost of I/O.

Parallel Programming Environment

Work in the compiler area by Padua, Hoeflinger, Almasi, and Lin has focused on developing a new interprocedural analysis mechanism that combines dependence analysis, privatization, and reduction and induction analysis into a single pass. It is, to our knowledge, the first test capable of accurately analyzing loops involving non-affine subscript expressions like those found in FFT computations, as well as loops containing complex interleaved access. In addi-
tion, new techniques have been developed for parallelizing irregular and symbolic codes. The techniques for irregular codes hold the promise of compile-time parallelization for some loops containing subscripted-subscript accesses and cheaper run-time parallelization tests for others. The techniques for symbolic codes make use of the properties of special container objects like lists and stacks in languages such as Java or Fortran90. We also have developed a new technique for measuring the amount of locality that exists in programs. The metric has been used to estimate the number of cache misses caused by programs in the Perfect and SPEC95 benchmark suites. The estimate was found to be within 10% of the actual number of misses for those codes. Finally, new versions of traditional compiler optimization algorithms have been developed for analyzing explicitly parallel codes. Conventional compiler techniques may produce incorrect code when applied to parallel codes thereby forcing the programmer to turn off optimizations. Compiler techniques for explicitly parallel programs, therefore, are of great importance for correctness and performance.

The new interprocedural analysis mechanism was added to Polaris. This has led, as expected, to more accurate analysis and significant improvements in performance. In fact, interprocedural analysis of complete codes has now become possible due to a reduction of several orders of magnitude in compiler analysis time. Code also has been written to implement a new data forwarding and prefetching pass for Polaris, which shows a reduction in execution time of 60% for communication-intensive programs on 32 processors. A pass to generate multithreaded code has been implemented, with speedups ranging between 2.5 and 3.1 on four processors.

We plan to design an MPI-aware compiler that will be capable of accepting a Fortran90 program, then generating a program that contains double- or single-sided MPI communication, or shared-memory parallelism. The input program to this compiler could be sequential Fortran90 code, sequential Fortran90 with shared-memory parallelism directives, or Fortran90 code containing MPI calls. This compiler would be based on our existing Polaris compiler. The shared-memory parallelism directives we plan on generating are those for the SGI/Cray Origin2000 and the HP/Convex Exemplar, plus the OpenMP directives, which are becoming a standard in the industry. Such a compiler would give us the flexibility we need to produce efficient parallel code for our simulations on a wide range of target machines. This MPI-aware compiler will be used to help in the process of tuning the simulation codes for effective parallel execution.

One of the major challenges we face is developing a parallel programming methodology that will enable engineers and computer scientists with very diverse backgrounds to cooperate in the development of efficient parallel codes. We have discussed several strategies to organize the code and manage code development. As part of this work, the initial codes developed by CSAR application scientists for GEN0 that combine multiple computational domains (structures and fluids) as well as sample interface codes were studied. The program was transformed by hand to improve its performance (by improving its locality) and to remove race conditions detected during the hand analysis of the code.

Based on this early experience and on numerous discussions with engineering faculty, Kale, Bhandarkar, Ramachandran, Maiya, Wilmarth, and Yelon have developed a preliminary proposal for the structure of the parallel code for rocket simulation. The approach is based on multi-domain decomposition with clear separation of parallel and algorithmic logic and will be facilitated by object-based computing. The domain-specific modules, commu-
cation, and interpolation modules then can be implemented independently. This approach fits well with the evolving nature of the CSAR codes. Individual modules may use OpenMP to utilize SMP nodes effectively while others use explicit parallelism via message-passing or data-driven objects. This program structure requires support for multiparadigm interoperability. With this in view, we have ported Converse, a runtime system for parallel programming that supports multiparadigm interoperability, to ASCI machines such as ASCI-Red, Origin2000, and IBM SP. Charm++, a C++-based system for data-driven objects that utilizes Converse, also has been ported to these machines. Charm++ will be used as a base language to demonstrate ease of programming distributed data structures, implicit communication, and automatic object-based load balancing.

Several approaches to manage the tuning of the parallel code are under study. One of them calls for teams of engineers and computer scientists to work together in the development of each parallel module. In another strategy, computer scientists would take a clearly written code (although perhaps not very efficient) from engineers and proceed through automatic and manual transformations to produce a highly tuned code. Each approach has its advantages and disadvantages.

During the coming year, we will implement libraries and other mechanisms to support and demonstrate the co-existence of OpenMP with data-driven objects and MPI. We are in the process of building a demonstration program based on GEN0 code that uses object-based distributed data structures, and interface abstractions based on parallel objects. We will conduct experiments with this code to develop and substantiate our approach making it easier to build object-based complex simulation codes modularly without sacrificing efficiency. We will explore the issues in building distributed data structures with 3-D computational domains in next-generation codes. In order to make it more efficient and intuitive to program cluster of SMP nodes, we will add new features to Converse and Charm++. These features will allow building node-level abstractions in addition to current processor-level abstractions.

**Parallel Performance Analysis**

During the past year, Reed, Pantano, and Oly have continued the development of SvPablo, a language-independent graphical environment for instrumenting application source code and browsing dynamic performance data. The SvPablo browser provides a hierarchy of performance displays, ranging from color-coded routine and source code profiles to detailed information about a particular routine or a particular source code line, including per-processor metrics. One key feature of SvPablo is its language and architecture transparency, which is achieved by representing performance data via a meta-meta-format presentation of different events from different languages using the same graphical interface. Thus, with SvPablo, one can instrument and visualize HPF, C, Fortran 77, and Fortran 90 applications on various architectures. SvPablo performs run-time summarization, which allows performance measurements of programs that execute for hours or days on hundreds of processors. In addition, hardware performance monitoring is integrated with SvPablo, allowing the user to select a set of event counters to be monitored during run-time for presentation at a source line level. SvPablo allows users rapidly to identify and correct performance bottlenecks in their applications. In addition, we have continued to develop I/O instrumentation tools based on the Pablo.
We will continue working to develop a model for performance prediction and scalability analysis in the coming year. This model exploits both compiler-derived data on symbolic program variables and performance measurements from selected executions of the compiler generated code. The goal is to integrate this symbolic model with SvPablo creating an architecture and language-independent performance toolkit that will merge performance modeling, measurement, analysis, and prediction. This will allow application and system developers to explore the performance implications of software and hardware design choices, both for extant systems, hypothetical ones, and a combination of the two. Using this environment, one could determine how application performance changes with variations in the parallel system configuration or application problem size, and identify which code fragments will become the performance limiting bottlenecks as hardware or application parameters change.

3.6 System Integration

Group Leaders: Robert Fiedler, Dennis Parsons, and D. Scott Stewart

Faculty: Quinn Brewster, Philippe Geubelle, Michael Heath, Herman Krier, Dennis Parsons, and Scott Stewart

Technical Program Manager: Robert Fiedler

Research Scientists/Programmers: Amit Acharya, Prasad Alavilli, Thomas Jackson, Fady Najjar, John Norris, Danesh Tafti, and Shaojie Xu

Graduate Research Assistants: Jason Hales, Xiangmin Jiao, Alireza Namazifard, and Yuan Lin

Detailed computational simulation of rocket behavior requires the development of computer codes implementing mathematical models of each of the rocket components and the interactions among them. These components include the case, propellant, combustion layer, core gas, and nozzle. Associated with each component are a particular mathematical model, geometric configuration, discretization, and material properties. Interaction between two components takes place across the physical boundary between them. Not only are such boundaries potentially complex geometrically, but also in many cases the boundary between components changes in both position and shape with time. In a rocket, for example, the burning surface regresses as propellant is consumed.

![Diagram](image)

Fig. 3.14 System integration enables increasing complexity in rocket simulations.
System integration involves two major tasks to ensure the physical, mathematical, geometric, numerical, and software compatibility of the component models and the codes implementing them. The first major task is providing information transfer across component boundaries. Boundary conditions for the component models must be compatible mathematically (e.g., an outflow from one component becomes an inflow for a neighboring component). The discretizations of neighboring components must fit together geometrically. Different spatial resolutions and discretization methodologies must be reconciled via interpolation where necessary. The other major task is temporal coupling of the components so that the whole system is evolved in a self-consistent manner. Different components may have very different time step sizes due to the choice(s) of algorithm(s) (e.g., explicit vs. implicit methods), spatial resolution, stability restrictions, and/or the physics of the sub-problem that the module solves. The computational cost of forcing each module to take a time step determined by the module requiring the shortest step is likely to be prohibitive. We are investigating multiple strategies for coupling modules requiring different time step sizes while maintaining the accuracy of the overall simulation.

Our approach to system integration is to develop a main program and a collection of interface codes that perform the communication between each pair of components. We plan to follow an object-oriented design methodology that will hide the data structures and other internal details of the individual component codes. This will simplify development and maintenance of the interface codes and the component codes, as well as make it easier to swap different versions of the same component—a critical capability for determining the most efficient algorithms and implementations.

As the first major step toward the GEN1 system code, we have developed a fully coupled, 3-D rocket simulation called GENH (H for half). GENH features:

- **Combustion**: mass is injected at a pressure-dependent rate and fixed temperature normal to the propellant surface. Since the burn time is short, we neglect regression of the propellant, but not its motion driven by the pressure of the fluid.

- **Fluids**: the fully compressible Euler equations are solved using ROCFLO, a shared-memory code developed by Alavilli and Tafti that uses body-fitted coordinates and is second order in time and space.

- **Structures**: the propellant is taken to be linearly elastic and compressible. A finite element method employing brick elements provided by Namazifard, Parsons, and Hales is employed to solve Cauchy’s equa-
tion of motion, given the pressure applied at the surface by the fluid. It takes implicit time steps, which can be much larger than the steps taken by the fluid solver.

The structure and fluid modules are coupled through an iterative scheme that predicts and then corrects each module’s solution after each time step in the solid until the convergence criteria are satisfied. Interpolation between the fluids and structures grids is simplified by constraining the two grids to coincide at the combustion interface. *Rocketeer*, a package under development by Norris based on the *Visualization Toolkit*, facilitates scientific visualization and animation of our simulations. Figure 3.16 shows output of the 3-D temperature field at early [left] and later [right] times.

![Rocketeer visualizations for the 3-D temperature simulation.](image)

**Fig. 3.16.** *Rocketeer* visualizations for the 3-D temperature simulation.