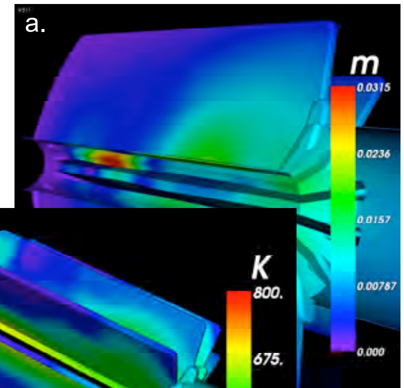
3 Research Program Accomplishments and Goals

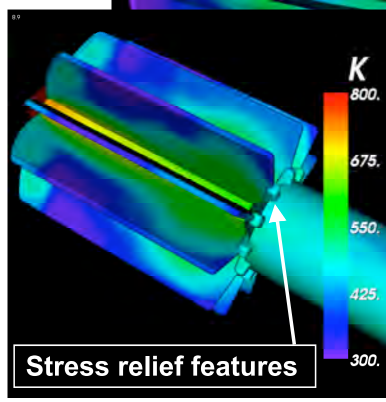
CSAR Legacy

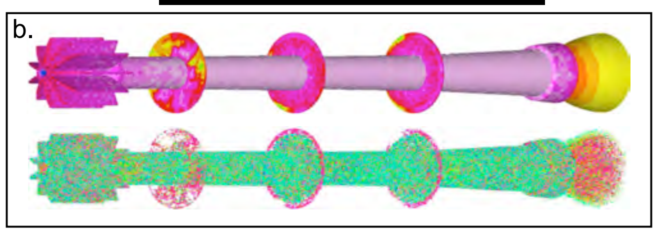
Perhaps CSAR's most obvious legacy is the *Rocstar* software suite and the fundamental research underlying it in turbulent multiphase flow, materials modeling, and propellant combustion, as well as its general integration framework for coupled multiphysics simulations. *Rocstar* is the nation's only integrated 3-D simulation capability for solid rocket motors, such as the Space Shuttle RSRM. An important current application, in collaboration with NASA and ATK/Thiokol, is using *Rocstar* to analyze the proposed five-segment enhancement of the RSRM for use in NASA's "Lunar Sooner" launch program.

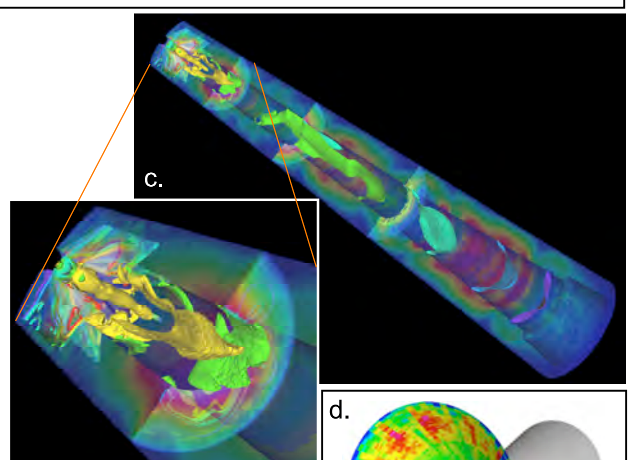
In addition to predicting overall performance of a given rocket motor design, simulations performed with *Rocstar* have also yielded unprecedented insights into many important details of solid rocket propulsion, including ignition transients, aeroelastic effects in joint slots and inhibitors, the role of heterogeneous combustion in seeding turbulence, slag accumulation around submerged nozzles, and ablative effects on nozzles due to particle impingement. CSAR's unique capabilities have been recognized by its participation as the only academic member of the Integrated Product Team for the USAF's current program for modeling and simulation in solid propulsion, as well as through funded research collaborations with ATK/Thiokol, Aerojet, Boeing, and Caterpillar.

Other lasting impacts of CSAR include its contribution to knowledge through its approximately 900 journal articles and conference papers, with numerous propulsion conference sessions devoted entirely to its research results; technology transfer both to and from NNSA national laboratories; the training of approximately 160 graduate students, 40 of whom have gone on to employment in DOE national laboratories; development of postdocs and other staff, five of whom have gone on to employment at DOE national laboratories; and career development of approximately forty faculty participants, which resulted in nu-









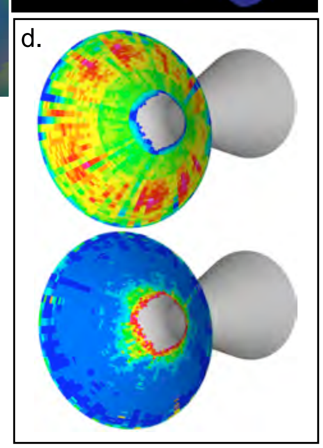


Fig. 3.0.1: *Rocstar* results from fully coupled, 3-D fluid/structure/combustion simulations of NASA Space Shuttle solid rocket booster shortly after ignition (a-c) and Air Force BATES motor (d). (a) Head end section shows local deformation of propellant due to pressure in core gas (top) and resultant surface temperatures (bottom). (b) Gas temperatures (top; red is hotter) and aluminum particle size and location (bottom; red is larger) in core flow. (c) Solid propellant deformation and temperature isosurfaces in fluid core. (d) Number of impacting aluminum droplets on rocket motor nozzle (top; red is higher) and mean diameter of droplets (bottom; red is larger).

merous promotions and awards, counter to the conventional wisdom about collaboration in academia, where individual accomplishments are usually most highly rewarded. Finally, although the Computational Science and Engineering (CSE) Program that hosts CSAR predated it, there is no question that CSAR substantially raised the visibility of CSE, both locally on campus and nationally, and contributed materially to its current status as a role model for developing interdisciplinary centers on our campus and similar programs elsewhere.

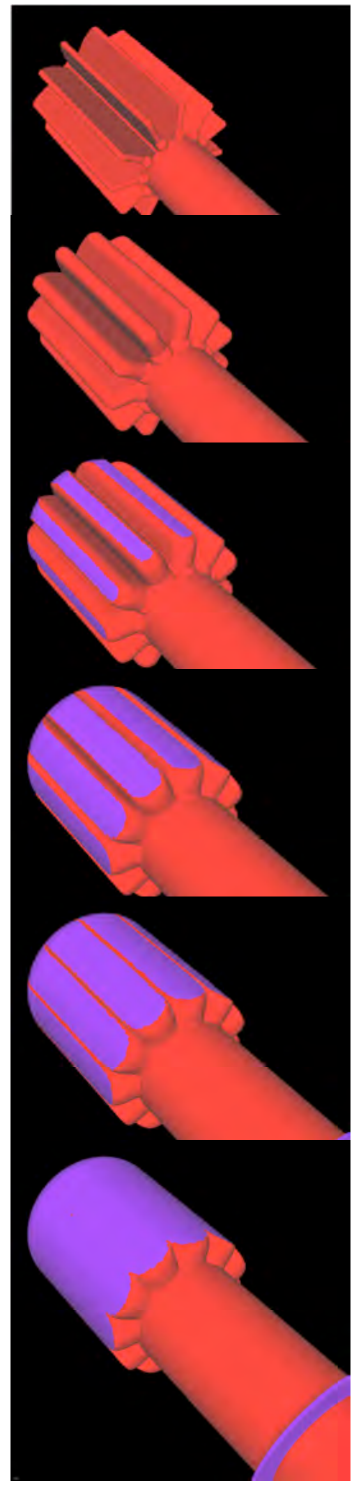


Fig 3.0.2: Complete propellant burnout in head end of RSRM (red is burning surface; blue is nonburning/extinguished.

Additional Research Accomplishment Highlights

- *Rocfire* — Provides detailed 3-D modeling and simulation of solid propellant combustion, enabling accurate prediction of propellant burn rate and turbulence injection into core flow. Based in turn on new, highly efficient parallel algorithms for generating realistic propellant packs.
- *Rocfluid-MP* — Unique multiphysics framework for computational fluid dynamics enabling simulation of the extremely complex turbulent multiphase flows in solid propulsion, based on new methods for multiphase flow, including the equilibrium Euler method for fine particles and Lagrangian superparticles for tracing larger particles, as well as both conventional and novel turbulence models, including optimal LES.
- *Roccom* — Software integration framework that enables independently developed physics modules to be easily integrated to perform coupled simulations, such as fluid-structure interactions in rocket propulsion.
- *Rocface* — New method for conservative data transfer between components that is several orders of magnitude more accurate than previous conservative methods. Conservation of mass, energy, and momentum is critical to maintaining fidelity of simulations over time.
- *Rocprop* — Implementation of a new stable and efficient method for explicit surface propagation, based on face offsetting, that is more accurate and efficient than implicit surface representation and enables detection and control of topological change, which are vital for multicomponent simulations with evolving geometries.
- Charm++ and Adaptive MPI — Provide scalable runtime support, including automatic load balancing, for large-scale parallel simulations while maintaining compatibility with MPI standard.
- Dynamic mesh adaptation capabilities, including mesh smoothing, mesh repair, and automated on-line remeshing. Crucial for maintaining mesh and solution quality over time for problem domains with evolving geometries.
- Other software tools for supporting large-scale parallel simulations, including the *Rocpanda* for parallel I/O, new tools for performance monitoring and prediction, custom 3-D visualization and animation, and a new hybrid geometric/topological mesh partitioner.

- Materials modeling — New multi-scale constitutive and damage models for heterogeneous solid propellants and metallic components that describe not only normal behavior, but also the initiation and evolution of material damage.
- Technology transfer to and from NNSA labs — Examples include polycrystalline plasticity model incorporated into LANL's MTS code, and SNL's Mesquite mesh smoothing tool parallelized and incorporated into CSAR's mesh adaptation strategy.

3.1 Program Overview

The central goal of CSAR is the detailed, whole-system simulation of solid propellant rockets under normal and abnormal operating conditions. Full simulations (Figure 3.1.1) 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 (Figure 3.1.2).

Our approach to system integration has been to develop a single executable code containing modules for the various components and the interface code for tying them together. We have followed an object-oriented design methodology that hides the data structures and other internal details of the individual component codes. This simplifies development and maintenance of the interface code and the component codes, and also makes it easier to swap different versions of the same component—a critical capability for determining the most efficient algorithms and implementations.

CSAR has evolved a series of increasingly sophisticated computational models for the primary rocket components and their interactions (Figure 3.1.2). This year's efforts were devoted to integrating newly developed physics models into *Rocstar*, verifying and validating the coupled code, performing many small- and large-scale simulations, and implementing and exercising *Rocstar* 3,

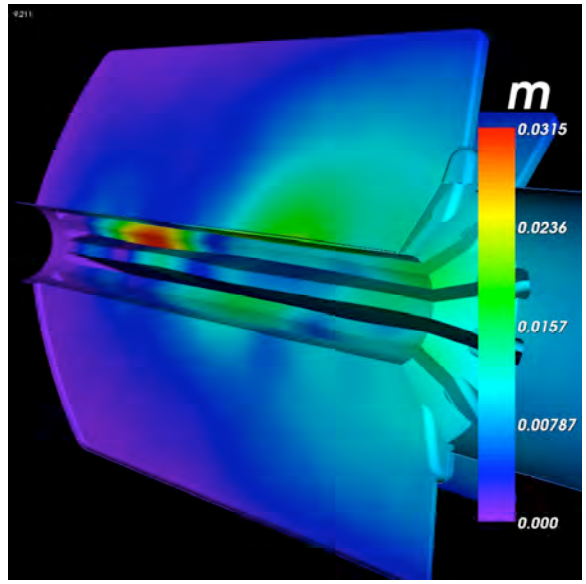
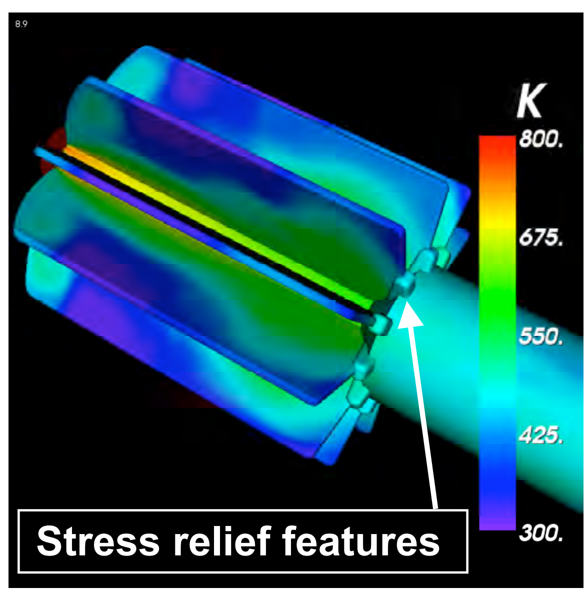


Fig. 3.1.1: *Rocstar* results from 3-D simulation of NASA Space Shuttle solid rocket booster shortly after igniter is triggered. Head end section (above) shows local deformation of propellant due to pressure in core gas; (below) resultant surface temperatures. Simulation employs igniter gas evolution, flame-spreading model, dynamic burning rate calculation for propellant regression, micromechanics-based propellant constitutive response model, and chamber gas model. Simulation purpose was to elucidate details of flame spreading and to determine structural response to internal gas pressure loads. Validation included comparison of head-end pressure history and axial quasi-steady operating pressure.



which has added powerful new capabilities including advanced mesh modification schemes.

Our initial implementation (GEN1) of the integrated simulation code *Rocstar* was operational at the end of 2000. It provided a simplified characterization of various burn scenarios. The GEN1 code employed macroscopic models for the separate components to enable a strong 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, constituted the key features in the second-generation code (GEN2), developed during FY01 and FY02. The refined models reflected the synthesis of fundamental, subscale studies, which are critical for detailed simulations of accident scenarios and for reliable simulation of multiscale phenomena such as combustion and turbulence. *Rocstar* 2.5 was a fully-functional motor simulation code that served as a strong foundation for future releases.

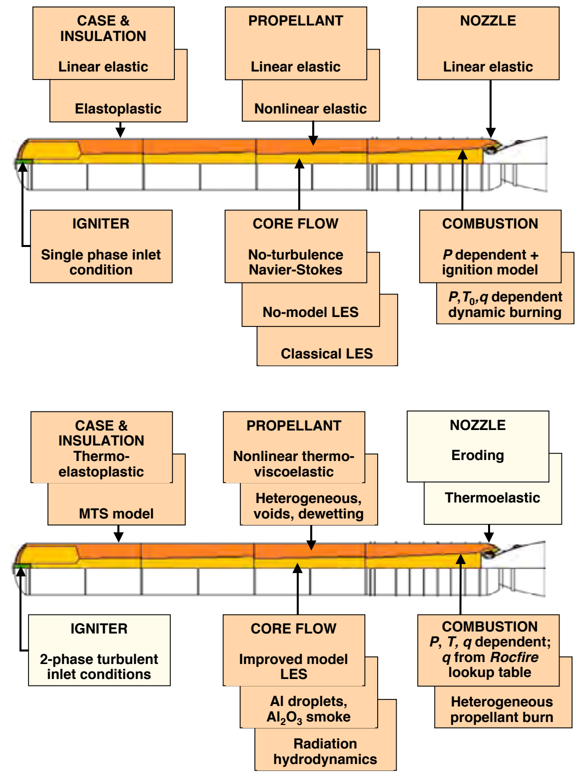


Fig. 3.1.2: Module development for *Rocstar* follows path of increasing complexity of physical component models. *Rocstar* enables users to select from among a spectrum of integrated computational models and algorithms.

The computer science integration efforts define the framework for these interconnections and, consequently, their impact on overall code performance. 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 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.

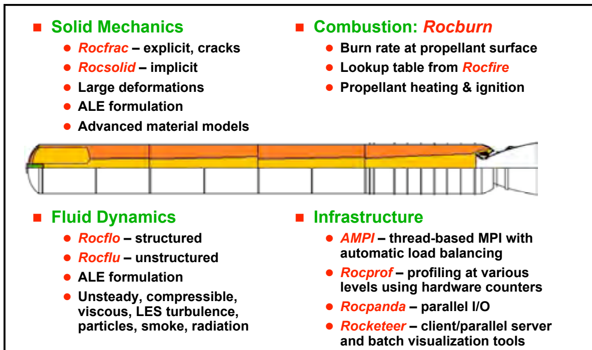


Fig. 3.1.3: Key component modules of *Rocstar* integrate as single executable code.

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, and/or the physics of the subproblem 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 often prohibitive. We continue to investigate multiple strategies for coupling modules requiring different time step sizes while maintaining the accuracy of the overall simulation (Figure 3.1.3).