3 Research Program Accomplishments and Goals

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 entail 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).

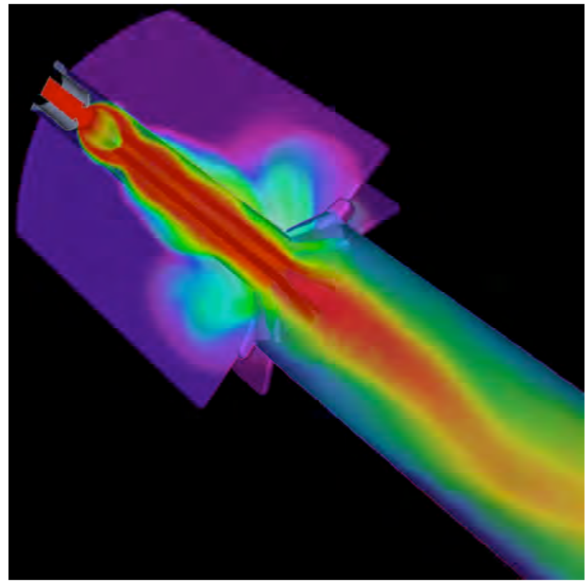

Fig. 3.1.1: *Rocstar* results from 3-D simulation of Space Shuttle solid rocket booster 52 milliseconds after igniter is triggered. Propellant and gas domains are cut in half lengthwise to show cross section of interior of head-end segment. Igniter is gray and red object at upper left. Remainder of visible surface represents the propellant, whose geometry includes 11 slots (4 are visible) that increase burning surface area (and therefore thrust) at liftoff. Further down, geometry changes to cylindrical bore. Colors (purple - magenta - blue - cyan) indicate surface temperature from 350 to 500 degrees K. Also shown is gas temperature, represented as translucent surface (to allow geometric details to show through), whose colors (green - yellow - red) correspond to range 300 K to 1700 K.

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 GEN2.5 of *Rocstar*, verifying and validating the coupled code, performing several large-scale simulations, and designing and implementing *Rocstar* 3, which adds 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, constitute 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.

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.

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).

Table 3.1.1 highlights the main accomplishments for FY04 for each of the CSAR Research Groups. Sections 3.2 through 3.6 provide additional details.

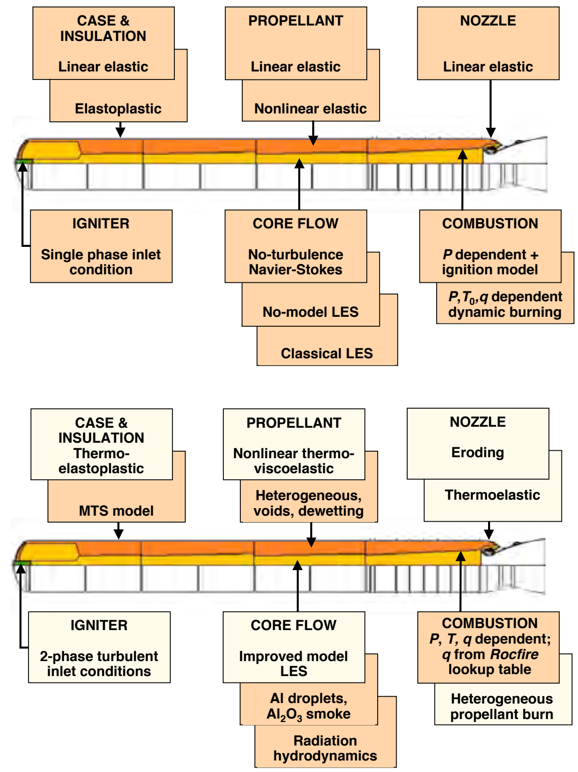

Fig. 3.1.2: Module development for *Rocstar* follows path of increasing complexity of physical component models. *Rocstar* enables users to select from among a spectrum of integrated computational models and algorithms.

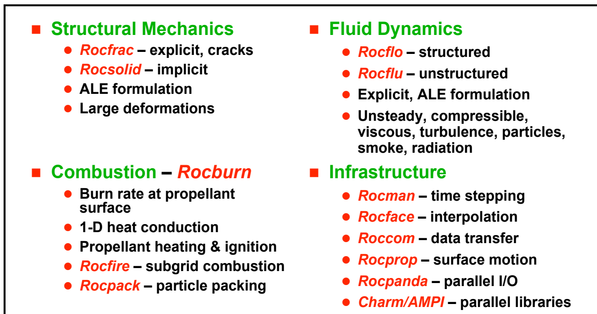

Fig. 3.1.3: Key component modules of *Rocstar* integrate as single executable code.

Table 3.1.1: Key Thrust-level Tasks and Accomplishments for FY04

Combustion and Energetic Materials (Section 3.2)

Local radiant ignition process simulated and verified
New three-step kinetic model developed for combustion processes in gas phase
Level-set technology employed to explore complex surfaces arising from aluminum particles in SP
3-D combustion and flame spread in SP cracks modeled
Constitutive modeling of solid propellant from first principles continued
Rational_Rocburn implemented to model unsteady combustion effects at propellant time scale
Ignition map developed for AP-composite propellants
Complete motor burnout simulation was achieved using *AXS/WaveTracker*
New ultrafine aluminum combustion study initiated

Computer Science (Section 3.3)

Support for dynamic arrays added to *Roccom*
Rocin/Rocout developed to enable mapping between *Roccom* and commercial file formats (HDF, CGNS, etc.)
Mesh optimization module, *Rocmop,* implemented that integrates CSAR-parallelized *Mesquite* from Sandia
Automated and intelligent mesh partitioning algorithm developed
Performance and mesh quality measures added to *Roccom* integration framework
Rocpanda parallel I/O enhanced with improved prefetching and caching strategies through GODIVA
ROMIO parallel I/O library enhanced in collaborative project with Argonne National Lab
New performance profiling tool, *Rocprof,* developed and implemented
AdaptiveMPI (AMPI) improved through enhanced support for checkpointing and fault tolerance
Confirmed methods for porting CSAR codes to BluGene/L
Performance and scalability studies continued on local and ASC machines

Fluid Dynamics (Section 3.4)

Incorporated second-order spatial discretization in *Rocflu*
Rocspecies developed and implemented
Conservative random ejection model implemented in *Rocpart* to simulate random ejection of Al particles from surface elements
New "time-zooming" algorithms developed
LES turbulence models used to study star-grain exit flows
Particle-turbulence interaction explored
New multiphysics interaction module (*Rocinteract*) developed and implemented for communication between fluids multiphysics modules
Turbulence modeling and experiments continued

Structures and Materials (Section 3.5)

New elements added to *Rocfrac* for enhanced bending and nearly-incompressible material behavior
Porous viscoelastic media material properties implemented in *Rocstar*
New stabilized mixed finite element added to *Rocsolid*
Coupling algorithms for fluid/structure interaction analyzed
Viscoelastic void growth model for solid propellant implemented in *Rocsolid*
Micromechanics-based particle dewetting model verified

System Integration (Section 3.6)

Rocstar 3 designed and initial coding complete
Improved single-processor performance of *Rocface* by factor of 15 on LLNL Frost machine
Employed new aluminum particle-laden motor design (BATES) in collaboration with rocket industry and U.S. Air Force
Implemented viscoelastic material model in *Rocstar*
Simulated fully-coupled simulation of turbulence around flexible inhibitor including LES
Performance evaluation, optimization, and algorithm characterization of *Rocstar* 2.5 completed
Rocstar 2.5 validated using Titan slumping and RSRM ignition transient simulations
Rocketeer visualization tool enhanced through modifications for CGNS data formats to incorporate other commercial meshing packages
Software engineering practices strengthened and codified
Developed and cataloged simulation problem sets