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 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, 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 simpli-

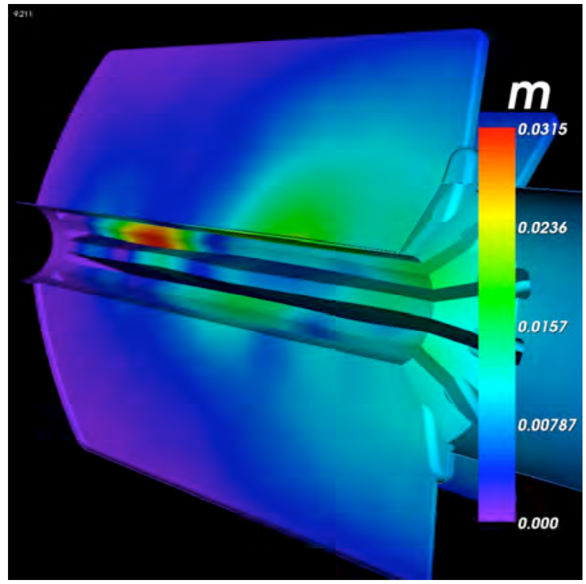
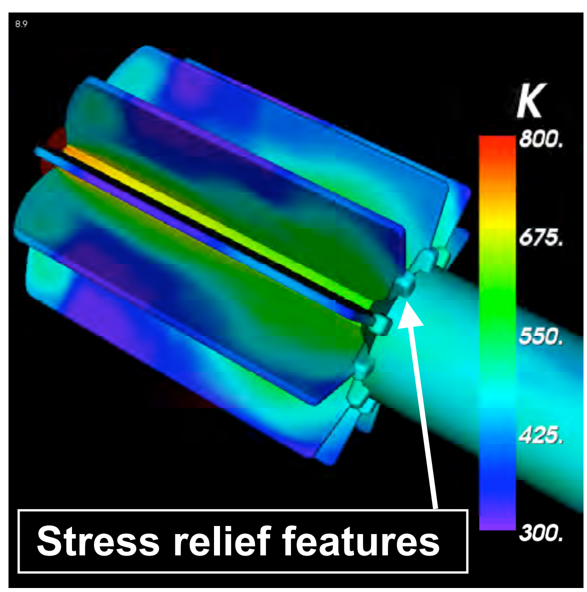


Fig. 3.1.1: *Rocstar* results from 3-D simulation of NASA Space Shuttle solid rocket booster shortly after igniter is triggered. Head end section (above) shows local deformation of propellant due to pressure in core gas; (below) resultant surface temperatures. Simulation employs igniter gas evolution, 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.



fied 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.

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).

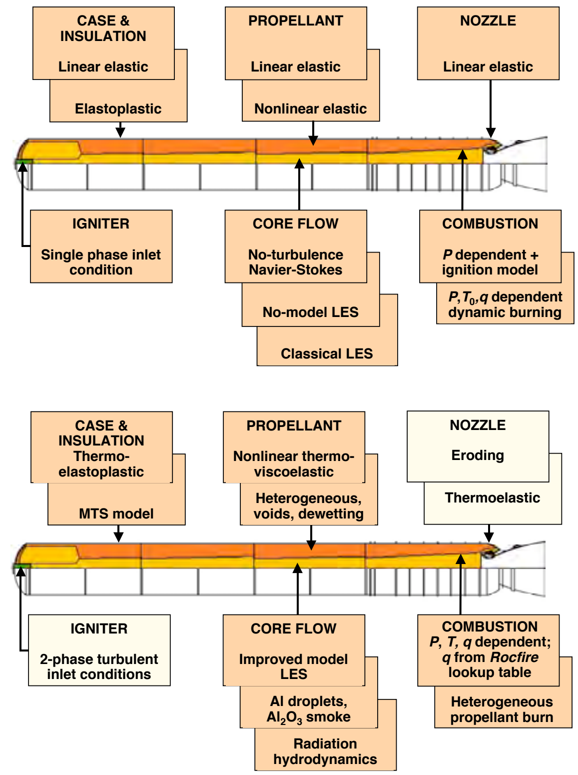


Fig. 3.1.2: Module development for *Rocstar* follows path of increasing complexity of physical component models. *Rocstar* enables users to select from among a spectrum of integrated computational models and algorithms.

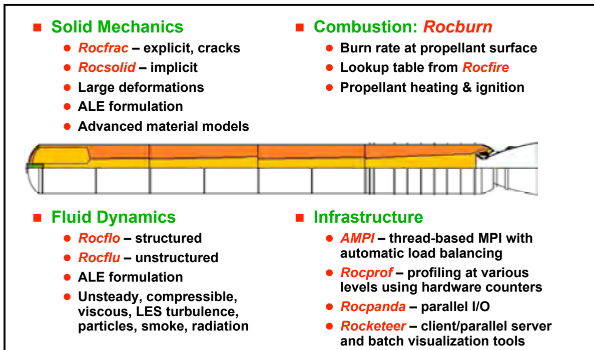


Fig. 3.1.3: Key component modules of *Rocstar* integrate as single executable code.

Table 3.1.1 highlights the main accomplishments for FY05 for each of the CSAR Research Groups. Sections 3.2 through 3.6 provide additional details.

Table 3.1.1: Key Thrust-level Tasks and Accomplishments for FY05

Combustion and Energetic Materials (Section 3.2)

Numerically examined impact of acoustic waves on burning heterogeneous propellant
Established proper injection boundary conditions for LES simulations
Predicted statistics for aluminum agglomerates on burning propellant surface
Developed lookup table for heat flux from surface from *Rocfire* simulations
Implemented user-friendly interface for *Rocpack* (known as *Rocprepack*)
Validated *Rocpack/Rocfire* in collaboration with Aerojet
Implemented *Rocfit* to calibrate reduced kinetic schemes
Instantaneous temperature field profiles above burning aluminized propellants simulated
3-D combustion and flame spread in SP cracks modeled
Flame structure in submillimeter-scale microburner simulated with *Rocstar*
UV/IR imaging spectroscopy used to obtain validation data for heterogeneous SP
Full-burnout 3D simulation completed including grain evolution and internal ballistics
Nanosized aluminum particle combustion process modeled
Effects of curved propellant surfaces on deflagration established
Developed theoretical framework for time zoomed grain burnback

Computer Science (Section 3.3)

Developed face-offsetting method to track interface regression accurately and stably
Implemented automated and intelligent mesh partitioning algorithm
Performed a systematic comparison of *Rocface* common-refinement method to other methods from literature
Designed and implemented a new indexing facility for *Rocketeer* that enables location of qualifying data values
Rocpanda updated to work with *Rocstar* 3
Developed three-phase approach to mesh adaptation based on severity of geometric change: mesh smoothing (*Rocmop*), local mesh repair, global remeshing (*Rocrem*)
Implemented *Rocrem* remeshing module; incorporated calls to Simmetrix
Parallel refinement and coarsening algorithms added to *ParFUM* (Parallel Framework for Unstructured Meshes)
Rocprof (profiling module in *Rocstar*) redesigned and adapted for use with *Rocstar* 3
Mesh optimization module, *Rocmop,* implemented that integrates CSAR-parallelized *Mesquite* from Sandia
Rocin/Rocout developed to enable mapping between *Roccom* and commercial file formats (HDF, CGNS 2.4, etc.)
Control and orchestration module, *Rocman*, redesigned and implemented with new coupling schemes
New hierarchical load balancing algorithm implemented for very large parallel machines (BGL, etc.)

Fluid Dynamics (Section 3.4)

Dual-timestepping (implicit) capability implemented in *Rocflo*
Many extensions in *Rocflu* implemented to match requirements of specific simulations (fully implicit Newton-Krylov solver, non-inertial frame of reference for flow over particles, new boundary conditions for jet flow resonance, improved crack combustion model, particle tracking, time zooming, etc.)
Development continued for incompressible solver in *Rocflu*
Enhanced performance of *Rocflu* (30% single-processor, 30-40% on ALC cluster)
Al particle breakup model based on critical Weber number implemented in *Rocpart*
Ejection model implemented for aluminum droplets from propellant surface

Combustion model including Al oxide vapor energy formalism implemented in *Rocpart* to enable full heat release in core flow

Established that freestream turbulence has little systemic effect on time-averaged mean drag on entrained particles

Developed two new models for LES of particle-laden turbulent flows (LES-equilibrium model and two-way-equilibrium model)

Improved and implemented a time-zooming formulation in *Rocflu* that showed speedup of 20-50 times

Structures and Materials (Section 3.5)

Added cohesive FE *Rocfrac* for nonmatching meshes

Incorporated new viscoelastic material constitutive model in *Rocsolid*

Developed and implemented material model in *Rocsolid* to account for damage evolution due to porosity

Explored several coupling schemes to better understand stability characteristics

Developed mixed stabilized finite element formulation based on Hughes Variational Multiscale principle

Continued development of axisymmetric finite element capable of non-axisymmetric loadings

Studied flow characteristics around flexible cylinders

Discovered crack-tip velocity singularity for dynamic cohesive crack growth

Implemented unified software architecture to facilitate Multiphysics simulations in space-time DG framework

Formulated 3D apparatus for nearly-incompressible solids subjected to finite deformation

Derived novel multiscale cohesive modeling framework incorporating physical processes in adhesive/interfacial layer

Developed new micromechanics model of particle dewetting in energetic materials

Implemented in *Rocsolid* nonlinear finite-strain constitutive model for SP under continuously evolving microstructure due to particle dewetting

Performed constant strain rate cyclic deformation simulations at the atomic level

Applied mesoscale models to rate effects in fracture, including dynamic strain aging

System Integration and Simulation (Section 3.6)

Completed *Rocstar* 3 implementation and verification

Rewrote and implemented all preprocessors to utilize *Rocin/Rocout*

Simulated RSRM ignition transients using *Rocflu, Rocfrac, and RocburnPY*

Demonstrated that convection occurs rapidly in star slots, but not in joints slots nor submerged nozzle

Improved 3D simulations of slumping in Titan IV field joint slots

Incorporated new nonlinear viscoelastic propellant models that include effects of voids and particle dewetting

Performed fully-coupled simulations of gas flow in vicinity of flexible inhibitors

Used superseismic shock as validation of fully-coupled fluid-structure interaction problem

Developed and implemented four new validation problems: superseismic shock; elestic piston; mesh motion in a tactical SP motor; and flexible panel in shock tube

Continued improvement of software engineering initiatives

Automated nightly regression testing through development and implementation of *Rocbuild, Roctest* and *Rocdiff*

Rocprep implemented to automate preprocessing of *Rocstar* 3 datasets