Center for Simulation of Advanced Rockets

University of Illinois at Urbana-Champaign

The primary goal of CSAR is the detailed, whole-system simulation of solid propellant rockets under both normal and abnormal operating conditions. We have implemented a highly scalable, fully integrated, three-dimensional code and calculated the coupled gas dynamics and structural mechanics of the Space Shuttle booster from ignition to the onset of steady burn. Refined multiscale component models and an advanced system integration framework based on experience gained from the current code are under development for our second generation virtual prototyping tool for solid propellant rockets.

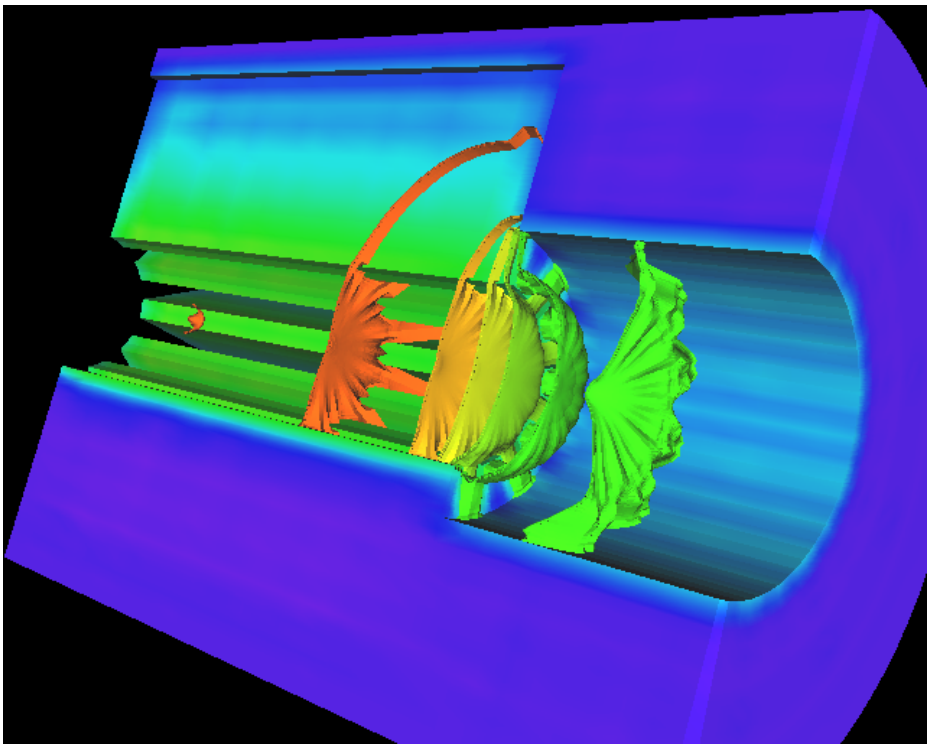

Figure 1. Simulation of Space Shuttle Booster 30 ms after ignition. Computational volume is truncated 5.7 m from head end and cut in half lengthwise to show interior cavity. Average stress is shown at propellant surface, ranging from –5 (magenta) to 34 (red) atmospheres. Gas pressure is represented by isosurfaces ranging from 1 (magenta) to 39 (red) atmospheres.

Solid Propellant Rocket Simulation

Solid rocket boosters (SRBs) are the “heavy lifters” of the space launch industry. Most of the world’s large, multistage launch vehicles—including the Ariane, Delta, Titan, and Space Shuttle—employ two or more SRBs to provide 80 percent or more of the thrust needed to lift a large payload off the launch pad and propel it the first few tens of miles above Earth.

Safety and reliability remain paramount concerns in rocket motor design because of the enormous expense of rockets and sophisticated payloads. In the spring of 1999, for example, a series of three consecutive launch failures collectively cost more than $3.5B.

The use of detailed computational simulation in virtual prototyping of products and devices has had enormous impact in some industries, for example in the design of automobiles and aircraft, but to date this approach has not made significant inroads in rocket motor design. Although simulation may never completely replace empirical methods, it can potentially dramatically reduce the cost of empirical methods by identifying the most promising approaches in advance of building actual hardware.

Simulating SRBs presents many challenges. Their complex behavior requires fully three-dimensional modeling to capture the essential physics adequately. Examples include the combustion of composite energetic materials; the turbulent, reactive, multiphase fluid flows in the core and nozzle; the global structural response of the propellant, case, liner, and nozzle; and potential accident scenarios such as pressurized crack propagation, slag ejection, and propellant detonation. The coupling between components is strong and nonlinear. The geometry is complex and changes dynamically as propellant is consumed. The spatial and temporal scales are extremely diverse.

Modeling and simulating a coupled system is even more demanding, as it requires not only still greater computational capacity, but also that the corresponding

software modules interact in a manner that is physically, mathematically, and numerically correct and consistent. When data are transferred between components, physical conservation laws must be honored, mathematical boundary conditions must be mutually satisfied, and numerical accuracy must be preserved, even though the corresponding meshes may differ in structure, resolution, and discretization methodology.

The enormous computational capacity required for integrated, whole-system simulation of an SRB is currently available only through massively parallel systems. Thus, the software integration framework, mesh generation, numerical algorithms, I/O, and visualization tools necessary to support such simulations must be scalable to many processors.

GEN1 Code

Our first generation rocket simulation code (GEN1) consists of a 3-D fluid dynamics solver (ROCFLO), a 3-D structural mechanics code (ROCSOLID), and an interface code (ROCFACE) for mesh association and interpolation.

ROCFLO solves the Navier-Stokes equations in an Arbitrary Lagrangian Eulerian (ALE) formulation, which enables its block structured mesh to move with the propellant surface. This Fortran 90/MPI module uses a second-order upwind TVD finite volume scheme with explicit time stepping. The initial fluids mesh is generated using GridGen.

ROCSOLID is a finite element code with an unstructured hexahedral mesh, a multigrid equation solver and implicit time stepping. The Fortran 90/MPI algorithm has recently been extended to an ALE formulation to allow for regression of the propellant. The initial structures mesh is generated using TrueGrid and partitioned for parallel execution using METIS.

ROCFACE determines which element on the structures surface is associated with each point on the boundary of the fluids mesh. This C++/MPI code also performs interpolation between meshes in a conservative manner.

The fluids and structures modules are temporally coupled using a “partitioned” or “operator-split” scheme. First, all processors execute the fluids module to take typically ten explicit time steps using an estimate of the location and velocity of the structures surface. The new values of the gas pressure and velocity at the fluid surface are converted to tractions and transferred to the structures module via the interface code. Next, all processors execute the structures module to take an implicit time step to catch up with the fluids module. The new location and velocity of the structures surface are transferred to the fluids module, which compares them to the estimated values it had used previously to reach the new time level. If these two sets of interface conditions differ by more than a specified amount, the same fluids and structures time steps are repeated using the new estimates. Iterating to consistency in this manner ensures that the coupled solution is as accurate as the least accurate module. We have found that convergence is typically attained after only one or two iterations.

Space Shuttle Booster Simulation

Our first large simulation of the Space Shuttle booster was run on a 256-processor SGI Origin 2000. There were roughly 4 million fluid cells and 300,000 structural elements. The case was taken to be rigid, while the propellant was linear elastic. The propellant surface was assumed to ignite instantaneously, acting as a moving mass injection boundary condition for the fluid. Regression due to burning of the propellant was neglected because of the short burn time. We followed the evolution for 100 ms, taking over 30,000 time steps during ten days of wall clock time.

We developed Rocketeer to visualize our results (see Figure 1). This C++ application is based on the Visualization Toolkit, which uses OpenGL. Rocketeer runs on Windows and Unix platforms, and intelligently reads and seamlessly merges HDF data files. It supports multiblock structured and unstructured meshes, and automates the generation of animation frames.

GEN1 Enhancements

We are improving our rocket simulations in many ways. For instance, we are adding a large eddy simulation turbulence model to ROCFLO. We are also including a simple ignition model to follow the heating of the propellant surface (initially due to the igniter inflow) until it reaches the critical temperature and starts burning. With this enhancement, we expect much better agreement between the computed and measured pressure history during the ignition transient. Our earlier calculations have already predicted a head end pressure at the onset of steady burning that was within ten percent of the measured value for the Shuttle booster.

ROCSOLID is being extended for large deformations in order to simulate an accident scenario involving propellant slumping in the Titan IV prequalification motor test. The case can be treated as linear elastic, and a viscoelastic model for the propellant is under development.

Parallel I/O using the PANDA library has been added so that data sets that would have been written to separate files for each processor are merged and written to a single file by each I/O node. These files can be transferred automatically to graphics workstations and tertiary storage by the I/O nodes as the simulation continues unimpeded on the compute nodes.

GEN2 Research

Our planned second generation solid rocket virtual prototyping tool (GEN2) requires research and development efforts in several relevant areas of computer science. These include parallel programming environments, performance monitoring and evaluation, linear solvers, mesh generation and adaptation, surface tracking, and visualization.

The GEN2 code will utilize an adaptive software framework for component integration based on decomposition and encapsulation via objects. This parallel programming environment provides automatic adaptive load balancing in response to dynamic change or refinement, as well as high-level control of parallel components. It is purposely designed to maintain compatibility with component modules in conventional languages such as Fortran 90, and it provides an automated migration path for our existing parallel MPI code base. This work is in part an extension of the previously developed Charm system, which has been notably successful in building a large parallel code for molecular dynamics.

Rocketeer will be upgraded to a client/server model to speed up visualization of remote data sets. The server part will run in parallel on a large multiprocessor at the remote site, while the client will control graphics operations and render images on a local desktop graphics workstation.

The individual GEN2 components will be improved based on detailed subscale simulations of phenomena such as propellant flames, pressure-driven crack propagation, turbulent flows with particles, and constitutive models for materials.

For more information contact — Robert A. Fiedler (rfiedler@uiuc.edu).

The CSAR research program is supported by the U. S. Department of Energy through the University of California under subcontract B341494.