41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit
10-13 July 2005, Tucson, AZ

AIAA 2005-3993

3-D Simulations of Ignition Transients in the RSRM

R. A. Fiedler*, A. Haselbacher†, M. S. Breitenfeld‡, P. Alexander§, L. Massa**, and W. C. Ross††
University of Illinois, Urbana, IL, 61801

We employ our tightly coupled fluid/structure/combustion simulation code ("*Rocstar* 3") for solid propellant rocket motors to study ignition, flame spreading, and the approach to steady operating conditions in the Space Shuttle booster (RSRM). A simple heat transfer model is used to compute local propellant heating due to the hot igniter gas flow until it ignites at the critical temperature. A 1-D dynamic burn rate model that takes transient behavior into account is applied to all cell faces on the propellant surface. We compare our pressure history to actual test firings. We also compare pressure values along the axis to measured data.

I. Introduction

IGNITION transients in the Space Shuttle Reusable Solid Rocket Motor (RSRM) have been studied for decades.[1-6] Although many of the older analyses used reduced geometries and empirically-based physical models, they were often able to match the pressure history of the RSRM to a fairly high degree of accuracy. However, reliance on calibration to fit experimental data greatly reduces the confidence one may have in the predictive capability of a simulation. More detailed multi-physics coupled simulations using the full 3-D geometry are required to understand all of the important phenomena taking place during the ignition of a large solid rocket motor.

The goal of the Center for Simulation of Advanced Rockets (CSAR) is to perform fully-coupled, 3-D, detailed simulations of solid propellant rockets using science-based models whenever possible[7]. This paper describes the Center's "*Rocstar* 3" code in considerable detail and presents results for RSRM ignition simulations. Verification and validation of *Rocstar* 3 is described elsewhere.[8] *Rocstar* 3 has recently been applied to a variety of other rocket-related problems, including propellant slumping in the Titan IV SRMU[9], flexible inhibitors[10], and multiphase flows in BATES motors[11].

II. *Rocstar* 3 Simulation Code

Rocstar 3 is the third generation integrated solid propellant rocket simulation package developed at CSAR. *Rocstar* 3 is a general-purpose solver for fully coupled, time-dependent fluid/structure/combustion interaction problems. It consists of a suite of physics applications coupled together by means of a powerful integration framework.[12] All components of *Rocstar* 3 are designed to run efficiently on massively parallel computers, enabling the use of detailed, science-based physical models in complex 3-D geometries.

A. *Rocstar* Architecture and Components

Figure 1 shows the basic architecture of *Rocstar* 3. A brief description of the specific modules that perform the functions written in each box is given below.

* Technical Prog. Mgr., Center for Simulation of Advanced Rockets, 2262 DCL, MC-278, 1304 W. Springfield Ave.
† Research Scientist, Center for Simulation of Advanced Rockets, 2253 DCL, MC-278, 1304 W. Springfield Ave.
‡ Research Prog., Center for Simulation of Advanced Rockets, 2233 DCL, MC-278, 1304 W. Springfield Ave.
§ Research Prog., Center for Simulation of Advanced Rockets, 3244 DCL, MC-278, 1304 W. Springfield Ave.
** Postdoc. Res. Asst., Center for Simulation of Advanced Rockets, 2234 DCL, MC-278, 1304 W. Springfield Ave.
†† Grad. Res. Asst., Center for Simulation of Advanced Rockets, 2265 DCL, MC-278, 1304 W. Springfield Ave.

1. Problem Set-up

On the left-hand side of Figure 1, the problem definition tools and the physics solvers are represented by blue boxes (with a lighter shade for the solvers). The selection of CAD packages is up to the user, as long as the package can output the geometrical information needed by the mesh generator(s). We typically employ Pro/Engineer to produce a CAD description of the fluid and solid domains, and export that information in IGES format. However, IGES is known for its lack of portability, and other formats may prove superior, provided the mesh generators can read them.

To some degree, the mesh generator may also be chosen by the user, although the physics application developers have written preprocessors that require mesh and boundary condition information in a very specific format. Our intention is to provide reader routines that support a number of commonly used mesh generators along with the preprocessors for the physics applications, which will allow the user to select any supported meshing tool. Currently, meshes and boundary conditions for the fluids codes are prepared using Gridgen, while meshes and boundary conditions for the structural mechanics codes are usually prepared using Patran or Truegrid, although it is possible to make some complete coupled input data sets using only Gridegen.

Once the meshes and boundary condition information are written in a supported format, the physics application preprocessors can be run either by hand or with the aid of the *Rocprep* input data set preparation tool. The preprocessors create complete input data sets partitioned for parallel execution for each physics application.

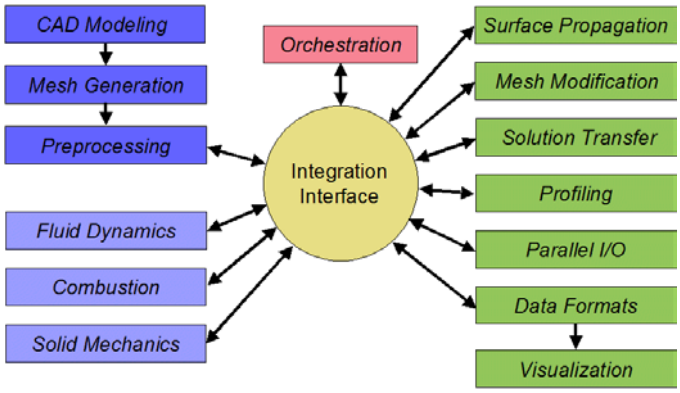

Figure 1. Rocstar 3 Architecture

2. Physics Applications

The 3 light blue boxes on the lower left in Figure 1 represent the various general-purpose physics solvers that are available for use with *Rocstar*. The existing fluid dynamics packages are called *Rocflu*[13] and *Rocflo*[14]. The basic algorithms in these codes were pioneered by Jameson.[15] *Rocflu* operates on unstructured tetrahedral or mixed tetrahedral/hexahedral/pyramid/wedge mesh cells to handle complex geometries, while providing high spatial resolution in boundary layers near physical surfaces. The fluid equations are formulated on moving meshes (Arbitrary Lagrangian Eulerian, or ALE scheme) to handle geometrical changes such as propellant burning and deformation. This cell-centered finite volume code employs a new high order WENO-like approach, as well as the HLLC[16] scheme to handle strong transient such as igniter flows, and a 3rd or 4th order explicit multistage Runge-Kutta time stepping algorithm. *Rocflo* uses either the central scheme or an upwind scheme based on Roe flux splitting[17] on multi-block structured meshes. In addition, *Rocflo* can use the Dual Time Stepping implicit algorithm (when applicable) to take time steps longer than the Courant (CFL) limit. Both fluid solvers can include turbulence (*Rocturb*)[18], Lagrangian superparticles (*Rocpart*)[11], smoke (*Rocsmoke*; equilibrium Eulerian method[19]), chemical reactions (*Rocspecies*), and radiation (*Rocrad*; diffusion approximation).

The rate of propellant deflagration is computed by one of three combustion modules. These physical models are one-dimensional (normal to the surface) in formulation, but are applied independently at each cell face on the burning propellant surface. The simplest model, *RocburnAPN*, adopts the well-known steady burn rate model in which the regression speed is proportional to the local gas pressure raised to some power "*n*". Two dynamic burn rate models may also be selected. Both solve a 1-D time-dependent heat conduction equation for the temperature profile in order to capture ignition transients. One of the dynamic models (*RocburnZN*[20]) is based on the Zeldovich-Novozhilov approach, while the other (*RocburnPY*) uses a simpler pyrolysis law. *RocburnPY* can also compute the heating of the propellant surface by hot igniter gases prior to burning, as well as ignition once the critical temperature is exceeded. A heat-flux look-up table computed by *Rocfire*, the detailed 3-D propellant combustion simulation code developed at CSAR, can be used by *RocburnPY* to determine the local instantaneous burn rate[21].

Rocstar includes two finite-element structural mechanics solvers, *Rocfrac* and *Rocsolid*.[22] Both solvers feature an ALE formulation to account for the conversion of solid propellant into the gas phase, handle large strains and rotations, can solve the 3-D heat conduction equation, and include a variety of element types and constitutive models. *Rocsolid* has an implicit time integration scheme that uses the multigrid method and/or BiCGSTAB to solve

the required linear systems efficiently in parallel. *Rocfrac* uses an explicit time integration scheme and can include cohesive volumetric finite elements between ordinary elements to follow crack propagation.

3. Integration Framework

The Integration Interface (center of Figure 1) is a library (API) called *Roccom*.[12] *Roccom* facilitates the exchange of data between different modules, including those written in different programming languages (C++, F90). By making a limited number of calls to *Roccom*, the physics applications gain access to a large number of useful components of our integration framework (column of boxes on the right-hand side of Figure 1).

The orchestration module controls the execution of the physics applications, including initialization, coupled time stepping, output dumps, and stopping criteria. The time stepping scheme is described below.

The surface propagation module (*Rocprop*) computes the motion of the propellant surface as it regresses due to burning. It can be switched off for problems in which there is no significant loss of mass from the solid domain (fluid/structure interaction without burning or evolution times << burn times). *Rocprop* includes a new, original, robust, and general surface propagation scheme called the face-offsetting method. *Rocprop* can be used in coupled simulations as well as fluids-only or solids-only calculations.

The mesh modification schemes in *Rocstar* are in various stages of development. Mesh smoothing (without changing the number of mesh vertices) for tetrahedral meshes is currently accomplished through calls to the mesquite package developed at Sandia National Lab[23]. Each partition calls mesquite concurrently, and then the vertices shared by multiple partitions are adjusted to make the smoothing process a parallel one. In the near future, local mesh repair will be performed by tools from Simmetrix, a company spun off from Professor Mark Shephard's group at Rensselaer Polytechnic Institute. We are also implementing an automatic remeshing module that uses the Simmetrix tools if and when that becomes necessary during a calculation.

The solution transfer module called *Rocface*[24] enables the physics applications to exchange interface quantities across non-matching meshes, which is essential to solving coupled problems. The interpolation scheme is exactly conservative by construction, because it operates on an overlay mesh, which is a common refinement of the two meshes on either side of the interface. Each subdivision of the overlay mesh lies entirely within a cell face in both surface meshes. Moreover, interpolation errors are minimized in the least squares sense, leading to a scheme that has been demonstrated to be 20 times more accurate than recently published methods.[25]

Rocstar automatically collects performance data for functions registered with *Roccom*, including physics application solution update times, data transfer times, output dump write times, etc. Profiling at the subroutine, loop, or statement level can be performed by inserting low-overhead calls to *Rocprof* in the source code.

Asynchronous Parallel I/O can be performed using *Rocpanda*. *Rocpanda* designates a user-specified number of processes as I/O servers, which collect data in the form of MPI messages from the compute processes, combine the data, and write it to disk in a manageable number of files in the desired format in the background as the simulation continues[26].

All major input and output by *Rocstar* is performed using *Rocin* and *Rocout*. These modules allow file-format independent I/O. The data file format to be used may be selected at run time without any changes required to the physics modules or their preprocessors. Currently HDF and CGNS[27] formats may be selected. Files in the latter format can be read by a number of third-party visualization tools, including *Rocketeer* (developed at CSAR).

4. Charm/AMPI

All modules in *Rocstar* use the MPI (Message Passing Interface) to pass messages between partitions. They are compatible with *AMPI*[28], an implementation of MPI developed at the University of Illinois that treats processes as user-level threads. There are two key benefits of *AMPI* for *Rocstar*: 1) the *AMPI* processes are "virtual" so that they can run on any number of actual CPUs, and 2) the virtual processes can be migrated from one CPU to another for dynamic load balancing. In performing large rocket simulations, we have used the first of these two features extensively to utilize available computational resources (few processors available than the number of partitions). For load balancing to help improve scalability, the load needs to be unbalanced by some change such as mesh adaptivity. We have only recently begun to add mesh adaptivity to *Rocstar*.

B. Coupled Time Stepping Schemes

In *Rocstar* we adopted the "partitioned" approach to time stepping, in which each domain (solid, fluid) is evolved separately from the other domains for one system time step. After each module reaches the advanced time level, it exchanges interface data with the other domains so that the system remains tightly coupled. The basic explicit time stepping scheme is depicted in Figure 2.

A system time step evolves the system from time level n (when the solution is known) to a new time level n+1. Currently, the size of the system time step is constant and chosen by the user. The time steps taken internally by explicit solid and fluid solvers are limited in size by the local CFL condition computed within those applications. If the system time step is larger than the CFL condition for a module, that module will take multiple internal time steps to reach the advanced system time level. We call these multiple internal steps "subcycles", although this terminology may have a different meaning in other contexts.

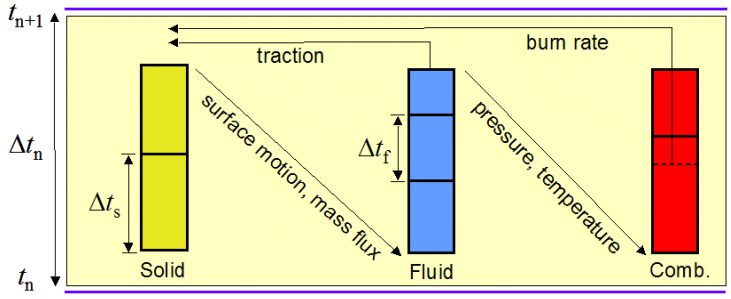

Figure 2. *Rocstar* 3 Coupled Time Stepping Scheme

In Figure 2, the system time step begins with the solid solver, which takes one or more internal steps to reach the advanced time level. To improve accuracy, an estimate (e.g., a linear extrapolation in time) of the load applied at the surface by the gas at the advanced time level may be used in this computation. When the solid solver reaches the new time level, the new surface location, velocity, and mass flux (due to burning) are passed to the fluid solver. (In practice, *Rocprop* actually moves the surface and determines the precise solid velocity and mass flux to use in the jump conditions at the burning surface. The implementation is designed to conserve mass while obeying Huygens' construction.) The fluid solver then advances the fluid solution to the new time level by taking one or more internal steps. The new load is passed to the solid, and the new surface pressure and temperature are passed to the combustion module, which determines the new burn rate and passes it to the solid. The new solution is now known at the new time level.

The accuracy of the above explicit time stepping scheme may be improved by repeating the computations required to advance from time level n to n+1, using the interface values at level n+1 that were obtained in the previous iteration as a better estimate of the burn rate and load on the solid surface at the new time level. We call such iterative improvement "Predictor-Corrector" cycles or iterations. The "Predictor" cycle is the same as the explicit method, while the "Corrector" cycles attempt to reduce the relative changes in the interface quantities from one iteration to the next to values below prescribed tolerances. P-C iterations are most useful when the implicit solid solver is selected for the simulation. For a detailed analysis of these time stepping schemes, see Ref. 29.

III. RSRM Problem Description

A. Geometry

The RSRM fluid domain (cut in half lengthwise along the axis) is shown in Figure 3. The model is much more detailed than that of our previous work. Our geometry includes the 11-point star grain region with curved star slots on the forward end and "stress relief features" on the aft ends of the star slot tips, a detailed model of the igniter (especially the opening into the main combustion chamber), 3 joint slots with inhibitors of the proper radii and correct propellant "overhangs", and the submerged nozzle. The fluid mesh we used consists entirely of tetrahedral elements (4.5 million nodes), although we have also generated 3 other meshes for this geometry, including 2 with mixes of

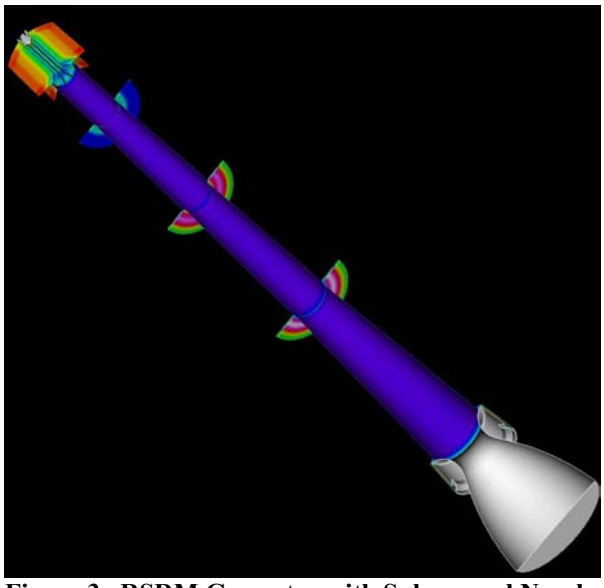

Figure 3. RSRM Geometry with Submerged Nozzle

tetrahedrons, pyramids, and hexahedrons. We have also generated a corresponding solid domain meshed with 3.5 million tetrahedral elements that is used to compute the structural response.

B. Material properties, gas properties

The gas is assumed to be ideal and inviscid with heat capacity at constant pressure 2552.8 J/kg-K. The ratio of specific heats is 1.13. The initial Pressure and temperature are 0.1 MPa and 300 K, respectively. The solid mass density is 1760 kg/m^3.

Heat from the igniter flow is transferred to the propellant according to a simple "film coefficient" law, which assumes that the heat flux is proportional to the temperature difference between the gas and the propellant. The film coefficient is 1500 W/m^2-K, which we obtained from 1-D simulations of RSRM ignition. Radiation is ignored. A science-based treatment of heat condition would require resolving the thermal boundary layer at the propellant surface, which would be prohibitively compute-intensive for an object of this size. The critical temperature for ignition is set to 850 K.

Our dynamic burn rate model reduces to the steady-state burn rate power law (with exponent n = 0.35) in the limit of slowly varying pressure (i.e., at quasi-steady operating conditions). When such a state is achieved, the propellant regresses 1.13 cm/s at a pressure of 68 atm.

C. Boundary Conditions

We model the fuel inside the igniter as a cylindrical surface through which hot gas is injected at a temperature of 2865 K. The time-dependent igniter mass flux is derived from Figure 11 of Ref. 3. The mass flux rises rapidly for the first 100 ms, drops by 30 percent over the next 250 ms, drops to 40 percent of peak at 500 ms, and then drops to 0 at 600 ms.

The propellant surface is initially treated as a flexible slip wall. Once it reaches ignition temperature, the propellant surface becomes a flexible mass injection boundary. The flame temperature is taken to be 2876 K. The usual jump conditions are used to conserve mass and momentum at the interface.

The outflow boundary (nozzle exit) is treated as a supersonic outflow ("continuation" or "zero gradient"). The inhibitors remain flexible slip walls.

On the outer (cylindrical) surface of the propellant, the case is taken to be rigid, as is the nozzle.

D. Initial Conditions

The gas is initially at rest at temperature 300 K and pressure .0.1 MPa. The propellant temperature is initially 300 K also.

IV. RSRM Results

We present results for a fluids-only simulation (the propellant is assumed to be rigid). A fully coupled simulation is in progress at the time of this writing. Figure 4 shows the head-end pressure history from our simulation, from the Space Shuttle Design Data Book (Thiokol Corp.), and from the TEM-6 test firing[30]. We note that there are significant differences between the Design Data Book's "nominal" behavior (presumably a prediction by a relatively simple analysis) and the actual test firing. These differences may indicate roughly the amount of variation that can be expected from one test firing to the next.

Our simulation exhibits many of the same features evident in the experimental data. After the igniter is triggered, there is a delay before the pressure begins to rise as the propellant is heated to the ignition temperature (the "induction interval"). The pressure subsequently rises are a rate quite similar to either the TEM-6 or the Design Book data. Our pressure rise is higher than it is in other simulations due to the full 3-D geometry and

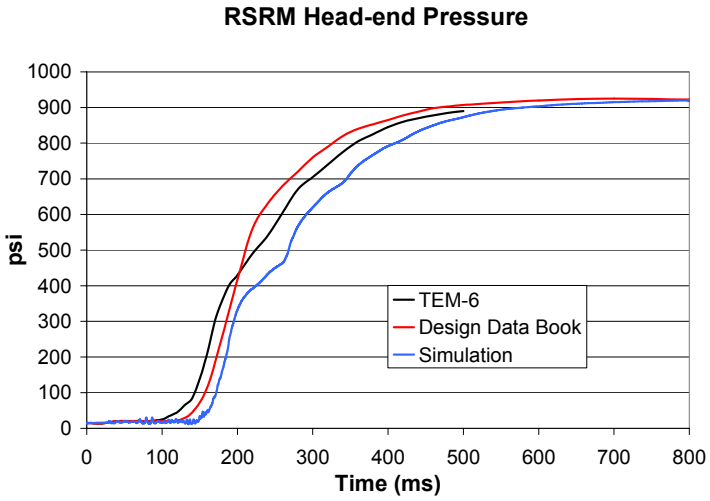

Figure 4. RSRM Pressure histories

our dynamic burn rate model. This is important because Shuttle specifications strictly limit the rate of pressure increase to protect the vehicle and its occupants.

At about 200 ms, there is a change in slope which corresponds to a rarefaction wave (superimposed on chamber filling, which causes a steady rise in pressure) moving away from the star grain region (where the propellant first begins to burn) as the front from the initial burst of hot gas from the propellant reflects from the head end and proceeds down the cylindrical bore of the rocket. After another 70 ms, the front has reflected from the nozzle and returned to the head end, causing the pressure slope to rise rapidly. The front reflects from the head end once again and the slope decreases. There is one more abrupt rise in pressure 70 ms after the one at about 270 ms. After that, the data and the simulations slowly converge to the same quasi-steady operating pressure.

The most obvious difference between our simulations and the data is that the delay between igniter triggering and initial pressurization is longer in the simulations, and this discrepancy becomes somewhat larger as the pressure increases. These differences can be explained by our omission of radiative propellant heating prior to burning. Radiation is expected to be the dominant source of heating in the joint slots and submerged nozzle region, where the hot igniter flow is impeded by cool air trapped in these rather narrow gaps. In the star grain region, flame spreading is quite rapid because the star slots are open at their aft ends and the igniter flow easily displaces the cool air initially present here.

Figure 5 shows the pressure as a function of position along the axis of the RSRM. The simulation results at 800 ms are included along with data at 1000 ms from two test firings[30]: QM-7, which was performed at 92 degrees F, and QM-8, which was performed at 39 degrees F. The simulation was run at 300 K or 80.6 degrees F.

All axial distances are normalized to the simulation data range. The experimental pressures are normalized to the simulation data range. The shifted experimental pressure data are normalized in the same manner, but then shifted so that the head end value equals 1. This helps us compare our results to pressure data for firings at different temperatures.

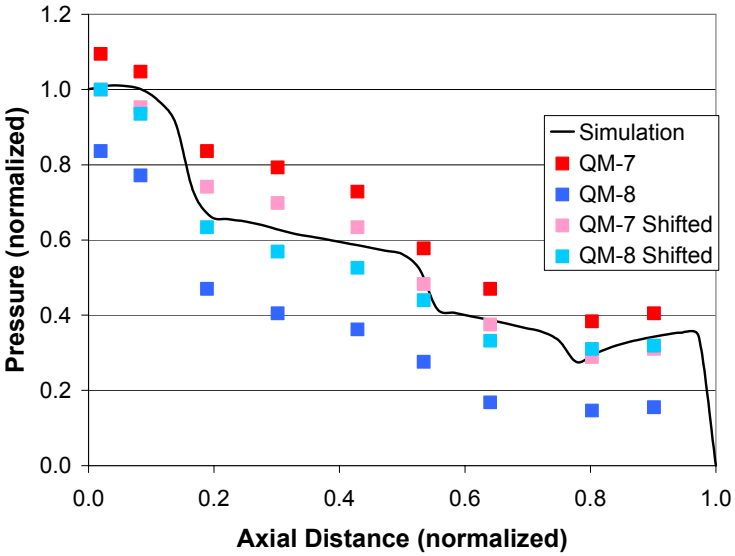

Figure 4. RSRM axial pressure profiles

Our simulation exhibits all of the major features seen in the experiments. The pressure drops slowly with position inside the star grain region (axial distance 0 to 0.1) and then drops steadily inside the cylindrical section of the head end segment. The pressure drops abruptly as we cross the forward joint (at 0.19). The pressure drops steadily inside the cylindrical forward center segment, and then drops abruptly again at the center joint (at 0.53). The pressure drops steadily again in the aft center segment, and drops abruptly at the aft joint (at 0.78). The pressure rises in the aft segment as the flow must turn to exit the nozzle. Finally, the pressure drops dramatically beyond the nozzle throat.

Given the difference between the two sets of experimental data, (even when shifted), it is not obvious how to achieve better agreement by improving our simulation, for example by including more physics such as turbulence. Another source of uncertainty is that our input parameters (burn rate, flame temperature, ratio of specific heats, etc.) may not closely match the conditions existing during either of the test firings.

V. Summary and Future Work

We described the Solid Propellant Rocket Simulation code developed at CSAR and applied it to study ignition transients in the Space Shuttle RSRM. In our simulation, heat transfer from the igniter gas to the propellant was computed using a simple film coefficient model. The pressure history and the axial pressure along the axis of the rocket compared favorably with experimental data. A coupled fluid/structure/combustion simulation using a new micromechanics-based constitutive model for the propellant is in progress.

Acknowledgments

The Center for Simulation of Advanced Rockets is supported by the U.S. Department of Energy through the University of California under subcontract number B341494.

References

[1]Caveny, L., Kuo, K., and Shackelford, B., "Thrust and Ignition Transients of the Space Shuttle Solid Rocket Motor", *Journal of Spacecraft and Rockets*, vol. 17, no. 6, 1980, pp. 489-494.

[2]Johnston, W., "Solid Rocket Motor Internal Flow During Ignition", *Journal of Propulsion and Power*, Vol. 11, no. 3, 1995, pp. 489-496.

[3]Luke, G., Eager, M., and Dwyer, H., "Ignition Transient Model for Large Aspect Ratio Solid Rocket Motors", AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA-1996-3273, 1996.

[4]LeHelly, P, "3D Turbuklent Navier-Stokes Simulations of Ignition Transients in Solid Rocket Motors", 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA-1998-3843, 1998.

[5]Salita, M., "Modern SRM Ignition Transient Modeling (Part 1): Introduction and Physical Models", 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2001-3443, 2001.

[6]Wang, J., "Modern SRM Ignition Transient Modeling (Part 5): Prospective Developments in CFD Simulation", 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2001-3447, 2001.

[7]Dick, W., Heath, T., Fiedler, R., and Brandyberry, M., "Advanced Simulation for Solid Propellant Rockets", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2005-3990, July 10-13, 2005.

[8]Brandyberry, M., Fiedler, R., and McLay, C., "Verification and Validation of the Rocstar 3-D Multi-physics Solid Rocket Motor Simulation Program", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2005-3992, July 10-13, 2005.

[9]Fiedler, R. A., Breitenfeld, M. S., Jiao, X., Haselbacher, A., Geubelle, P., Guoy, D., and Brandyberry, M., "Simulations of Slumping Propellant and Flexing Inhibitors in Solid Rocket Motors," 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2002-4341, July 11-14, 2002.

[10]Wasistho, B., Fiedler, R., Namazifard, A., and Brandyberry, M., "3-D Coupled Simulations of Flexible Inhibitors in the RSRM," 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2005-3996, July 10-13, 2005.

[11]Najjar, F. M., Massa, L., Fiedler, R., Haselbacher, A., Wasistho, B., Balachandar, S., and Moser, R. D., "Effects of Droplet Loading and Sizes in Aluminized BATES Motors: A Multiphysics Computational Analysis," 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2005-3997, July 10—13, 2005.

[12]Jiao, X., Zheng, G., Lawlor, O. S., Alexander, P. J., Campbell, M. T., Heath, M. T., and Fiedler, R. A., "An Integration Framework for Simulations of Solid Rocket Motors," 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2005-3991, July 10-13, 2005.

[13]Haselbacher, A., "A WENO Reconstruction Algorithm for Unstructured Grids Based on Explicit Stencil Construction", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA 2005-0879, July 10-13, 2005.

[14]Blazek, J., "Flow Simulation in Solid Rocket Motors Using AdvancedCFD," AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit ,AIAA-2003-5111, 2003.

[15]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[16]Batten P., Clarke N., Lambert C., and Causon D.M., "On the Choice of Wavespeeds for the HLLC Riemann Solver", *SIAM J. Sii. Stat. Comput.*, vol. 18, no. 6, 1996, pp. 1553-1570.

[17]Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," J. Computational Physics, vol. 43, 1981, pp. 357-372.

[18]Wasistho, B., and Moser, R. D., "Simulation Strategy of Turbulent Internal Flow in Solid Rocket Motor," Journal of Propulsion and Power, vol. 21, no. 2, 2005, pp. 251-263.

[19]Ferry, J., and Balachandar, S., "Equilibrium Expansion for the Eulerian Velocity of Small Particles", *Powder Technology*, vol. 125, 2002, pp. 131-139.

[20]Tang, K. C., and Brewster, M. Q., "Dynamic Combustion of AP Composite Propellants: Ignition Pressure Spike", AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2001-4502, 2001.

[21]Massa, L., Jackson, T. L., and Buckmaster, J., "Using Heterogeneous Propellant Burning Simulations as Subgrid Components of Rocket Simulations", *AIAA Journal*, vol. 42, no. 9, 2004, pp. 1889-1900.

[22]Namazifard, A., and Parsons, I. D., "A distributed memory parallel implementation of the multigrid method for solving three-dimensional implicit solid mechanics problems", *International Journal for Numerical Methods in Engineering*, vol. 61, 2004, pp. 1173-1208.

[23]Brewer, M., Freitag, L., Knupp, P., Leurent, T., and Melander, D., "The Mesquite Mesh Quality Improvement Toolkit", *Proceedings, 12th International Meshing Roundtable*, Sandia National Laboratories, Sept. 2003, pp.239-250.

[24]Jiao, X., and Heath, M. T., "Common-Refinement Based Data Transfer Between Nonmatching Meshes in Multiphysics Simulations", *International Journal for Numerical Methods in Engineering*, Vol. 61(14), 2004, pp. 2402-2427.

[25]Jaiman, R. K., Jiao, X. Geubelle, P. H., and Loth, E., "Assessment of Conservative LoadTransfer for Fluid-Solid Interface, with Nonmatching Meshes", *Int. J. Numer. Meth. Engng*., Vol. 0, 2004, pp. 1-45.

[26]Lee, J., Winslett, M., Ma, X., and Yu, S., "Tuning High-Performance Scientific Codes: The Use of Performance Models to Control Resource Usage During Data Migration and I/O", in *Proceedings of the 15th ACM International Conference on Supercomputing*, June 2001.

[27]Cosner, R., Oberkampf, W., Rumsey, C. Rahaim, C., Shih, T., "AIAA Committee on Standards for Computational Fluid Dynamics: Status and Plans", 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2005-568, Jan. 10-13, 2005.

[28]Huang, C., Lawlor, O., and Kale, L, "Adaptive MPI", in *Proceedings of the 16th International Workshop on Languages and Compilers for Parallel Computing*, College Station, TX, October 2003.

[29]Namazifard, A., Hjelmstad, K., Sofronis, P., Nakshatrala, K., Tortorelli D., and Fiedler, R. "Simulations of Propellant Slumping in the Titan IV SRMU Using Constitutive Models with Damage Evolution", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2005-3994, July 10-13, 2005.

[30]Laubacher, B., "Internal Flow Analysis of Large L/D Solid Rocket Motors", 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA-2003-2803, 2003.