

A Methodology for Actuating RCS Jets in a Continuous, Time-Accurate CFD Simulation

Alexandra Hickey¹, Victor Petitgenet², Bradford Robertson³, and
Dimitri Mavris⁴

Aerospace Systems Design Lab, Georgia Institute of Technology, Atlanta, GA 30332

Computational fluid dynamics simulations of Mars entry vehicle reaction control systems have been used to gain a better understanding of the wide range of flow phenomena encountered during various phases of flight. One aspect of this is simulating the system during controlled flight. Current practices model the system in a continuously actuated state, but this methodology is unable to accurately model the dynamic behavior of the system during EDL. One method to improve this is to create a coupled CFD-RBD-Control simulation. The first step to this is establishing a method of accurately actuating an RCS jet during a time accurate CFD simulation. This paper presents a method to achieve this actuation. There are three phases to the methodology: establishing the off state, creating a ramping function that will initialize the jet flow, and establishing the on state. A test case of a jet in a supersonic crossflow is implemented to establish the success of the methodology in being able to actuate a jet in the midst of an ongoing CFD simulation. It shows significant improvement in boundary condition responsiveness for linear and exponential ramping functions in comparison to the step function. A preliminary demonstration of the methodology is presented for the MSL vehicle which shows significant advantages for the implementation of a linear ramping function over a step function.

I. Nomenclature

L_{ref}^*	=	nondimensional reference length
$PR_{t,bc}$	=	nondimensional total pressure ratio at subsonic inlet boundary condition
$p_{static, freestream}$	=	nondimensional freestream static pressure
$p_{total, plenum}$	=	nondimensional total pressure at nozzle plenum
t	=	simulation step
t_{start}	=	simulation step at ramping function start
t_{end}	=	simulation step at ramping function end
$TR_{t,bc}$	=	nondimensional total temperature ratio at subsonic inlet boundary condition
$T_{static, freestream}$	=	nondimensional freestream static temperature
$T_{total, plenum}$	=	nondimensional total temperature at nozzle plenum
τ	=	scaling constant in exponential ramp function

¹ Senior Graduate Researcher, Aerospace Systems Design Lab, AIAA Student Member.

² Graduate Student, Aerospace Systems Design Lab, AIAA Student Member.

³ Research Engineer II, ASDL, School of Aerospace Engineering, Georgia Tech, AIAA Member

⁴ S.P. Langley Distinguished Regents Professor and Director of ASDL, Georgia Tech, AIAA Fellow

II. Introduction

As interest in exploring Mars continues to grow, the development of tools to allow for better modeling and design of entry, descent, and landing (EDL) vehicles is imperative. These vehicles most commonly use a blunt body design[1]. This design is favored due to its ability to dissipate the vehicle's kinetic energy during EDL. Aerodynamically, this shape has a strong bow shock and defined elongated wake. Monopropellant hydrazine thrusters are often used for the reaction control systems (RCS) to provide roll control during the entry mission phase. Interactions between the wake and the jet effluent of the RCS plays a critical role in the controllability of the vehicle[2]. These interactions significantly impact the pressure distribution and thermal environment of the backshell. The resulting changes to the aerodynamic forces have a considerable impact on the effectiveness of the vehicle's control system in certain flight regimes [3]. The accurate prediction of these interactions is therefore critical to the successful design of EDL vehicles.

Current CFD modeling of RCS systems on EDL vehicles relies on steady state and dynamic simulations where the jet is assumed to be on for the entire duration of the simulation. Additionally, the vehicle is modeled at a fixed attitude, speed, and environment. These simulations are then validated against experimental results from wind tunnel tests[3–6]. While these simulations and tests can provide some insight into the performance of RCS systems, they lack the ability to effectively model true dynamic behavior of the system as it descends through the atmosphere. These simulations encounter additional limitations due to the need for complex and extremely dense grids in the wake of the vehicle to capture the RCS wake interactions[2,3,7]. This creates a significant amount of uncertainty in the results that are derived from these simulations and ultimately led to the implementation of control deadbands on the Phoenix Lander[5,8]. Work by Brock et al. and Ernst et al. has taken a significant step forward in improving the modeling of the flight behavior of these vehicles in coupled CFD-rigid body dynamics(RBD) simulations and validating them against ballistic range tests[9,10]. Sahu et al. completed a preliminary integration of flight control systems(FCS) into a CFD-RBD framework to create a CFD-RBD-FCS simulation of the Army-Navy generic finner missile[11]. Their work represents a proof of concept that these models can be successfully integrated, but their simulations showed some numerical stability issues when actuating the jet via a step function which they resolved by implementing an unspecified ramping function during jet actuation.

This paper defines a detailed the methodology for actuating RCS jets within a time accurate CFD simulation and focuses specifically on the application of this method to an EDL vehicle. This expands upon the existing methodologies by focusing on the process of modeling the actuation of the jet for the off state, through its initialization, and to the on state. Each state is defined individually with its implementation described in detail. The methodology presented is then demonstrated on a physically meaningful test case, and initial results are shown on the Mars Science Laboratory (MSL) vehicle.

III. Methodology

The problem methodology is presented in three distinct phases: the off state, the initialization, and the on state. The off state represents the state in which the RCS jet is turned off and not providing any actuation; the initialization is the time between when the initial command to turn on the jet is sent, and the jet has fully responded to that command; and the on state is the state in which the jet plume has fully initialized. Each of these states is modeled as part of a piecewise function in the FUN3D Python API. The implementation is described below.

A. On state

The on state is the simplest to model and the first state that should be verified. In this state the boundary condition on the jet plenum is set to the subsonic inflow boundary condition in FUN3D. This boundary condition is controlled by temperature and pressure ratios defined in eq. (1) and (2). These ratios are defined by the total plenum state with respect to the static freestream state. Once the temperature and pressure ratio are determined, the on state should be run in a time accurate simulation to verify that the jet is behaving as expected. This is the state that should be used to determine the proper CFD settings and verify their validity since it is generally expected to have the most stable, complex flow behavior.

$$PR_{t,bc} = p_{total,plenum} / p_{static,freestream} \quad (1)$$

$$TR_{t,bc} = T_{total,plenum} / T_{static,freestream} \quad (2)$$

B. Off state

The off state covers the time from when the simulation starts to when the first command is sent to change the jet boundary condition. In an ideal situation this state would be modeled with an inviscid wall boundary condition. However, in FUN3D and many other CFD programs the type of boundary condition cannot be changed once set at the beginning of the simulation. This means that the subsonic inflow boundary condition that is suitable during jet actuation (the initialization and on states) must also be used for the off state. In order to use this boundary condition to define the off state, the ideal baseline must be established. This ideal baseline can be used to determine the temperature and pressure ratios used to define the subsonic inlet boundary condition for the simulated off state. Thus, the very first step is to run a time accurate simulation with an inviscid wall boundary condition. The simulation should be run long enough for the residuals to converge. Once this is complete, the results must be sampled to determine the off state boundary conditions.

In order to do this, slices are output from the flow field, and the last slice of the simulation, which should be taken at the last timestep, is loaded into Tecplot. Samples should be taken from as close to the boundary as possible. First a sample is taken at the left-hand side of the chamber, then the middle, then the right. These results may vary slightly but can be averaged together to establish the conditions at the boundary. Next eq. (1) and (2) are used to calculate the appropriate pressure and temperature ratios for the off state boundary condition. This off state should then be verified by running another time accurate simulation until it converges and inspecting the flow for significant differences from the ideal off state simulation.

C. Initialization

The initialization phase covers the modeling of the RCS response once it is actuated. Modeling the time-dependence of the jet inflow boundary condition is the primary focus of this paper. The pressure ratio is modeled with three different functions: a step function which follows eq. (3), a linear ramping function which follows eq. (4), and an exponential function which follows eq. (5). In each of the equations PR_{on} is the pressure ratio in the fully on state and PR_{off} is the pressure ratio in the off state. t , t_{start} , and t_{end} are defined as the current time step, the timestep at the start of the ramping function, and the time step at the end of the ramping function, respectively. τ in eq. (5) is a constant that is manually tuned for each problem such that the pressure ratio at t_{end} is nearly equal to the pressure ratio for the on state. The exponential function is expected to be most reflective of the pressure changes seen in a chamber during actuation[12,13].

$$\begin{aligned} PR_{t,bc}(t) &= PR_{off}; t < t_{start} \\ PR_{t,bc}(t) &= PR_{on}; t \geq t_{start} \end{aligned} \quad (3)$$

$$PR_{t,bc}(t) = (PR_{on} - PR_{off}) / (t_{end} - t_{start}) * (t - t_{start}) + PR_{off}; \quad t_{start} \leq t \leq t_{end} \quad (4)$$

$$PR_{t,bc}(t) = (PR_{on} - PR_{off}) * \left(1 - \exp \left[-\frac{t - t_{start}}{\tau} \right] \right) + PR_{off}; \quad t_{start} \leq t \leq t_{end} \quad (5)$$

Each of the above equations is coded as the jet initialization part of a piecewise function in the FUN3D Python API. This API allows users to change the pressure and temperature ratio values of the subsonic inflow boundary condition in Python. The initialization phase duration is ideally determined based on the response time of the system as determined through literature review, experiment, or manufacturer information. The response time may be shortened due to computational time requirements or estimated due to lack of availability of data. The temperature ratio is assumed to be step function defined by eq. (6) for all the simulations. The evaluation of different functions to model the change in temperature ratio could be the subject of future study but a step function was found to be an acceptable assumption for this demonstration. The implementation of such a study would follow the same methodology presented in this paper for the pressure ratio and modify the final ramping function to better capture the temperature changes seen in the chamber.

$$\begin{aligned} TR_{t,bc}(t) &= TR_{off}; t < t_{start} \\ TR_{t,bc}(t) &= TR_{on}; t \geq t_{start} \end{aligned} \quad (6)$$

IV. Test Case

Modeling the RCS system on an EDL vehicle requires a large and complex grid in order to capture the behavior of the jet, the wake, and their interactions. This has motivated the use of a test grid which isolates an RCS nozzle from the rest of the problem to allow for rapid testing of the methodology. The test case selected is a simple jet submerged in a flat plate and subjected to a crossflow which is a common demonstration case used to study RCS flow dynamics[14]. The flow phenomena for this test case are well understood for the fully initialized jet making it an ideal case to use for the initial methodology development and testing.

The implemented test case is based on a 400N Hydrazine RCS thruster [15]. The high-level design parameters of the system summarized in Table 1 were used to generate inputs to a nozzle design procedure described by Rao et al. [16]. This procedure approximates the nozzle contour for optimum thrust within prescribed length constraints. Quadratic Bezier curves are used to describe the bell of the nozzle. The throat and exit are defined by circular arcs as a function of the nozzle's throat radius.

Table 1: Hydrazine RCS thruster key characteristics [15].

Thrust Range	120-420 (N)
Nozzle Area Ratio	30
Propellant	Monopropellant Grade Hydrazine

The nozzle was then paired with a representative combustion chamber geometry for a catalyzed hydrazine monopropellant RCS system. This geometry was taken and revolved in SolidWorks to create the 3D geometry which was then embedded in a box where the floor of the box represented the flat plate, and the walls represented the boundaries of the grid. The grid boundaries were set to be at least 40 nozzle diameters in each direction based off of an estimation of the bounds of the jet effects from Dyakonov to ensure that all necessary flow phenomena were captured[3]. A 8 million node tetrahedral grid was produced based off this geometry in Capstone. The grid shown in Fig. 1 has boundary layers set up along the top of the flat plate and inside walls of the RCS jet. A couple of key sources had to be set up to control grid density to properly model the jet flow. The first is a very dense source over the nozzle exit. The second is a slightly less dense triangular source that starts slightly in front of the nozzle exit and extends midway back on the grid to capture the shock and other large-scale structures expected. A grid density study was completed to confirm sufficient density for the turbulence model and to ensure that all expected flow phenomenon could be captured.

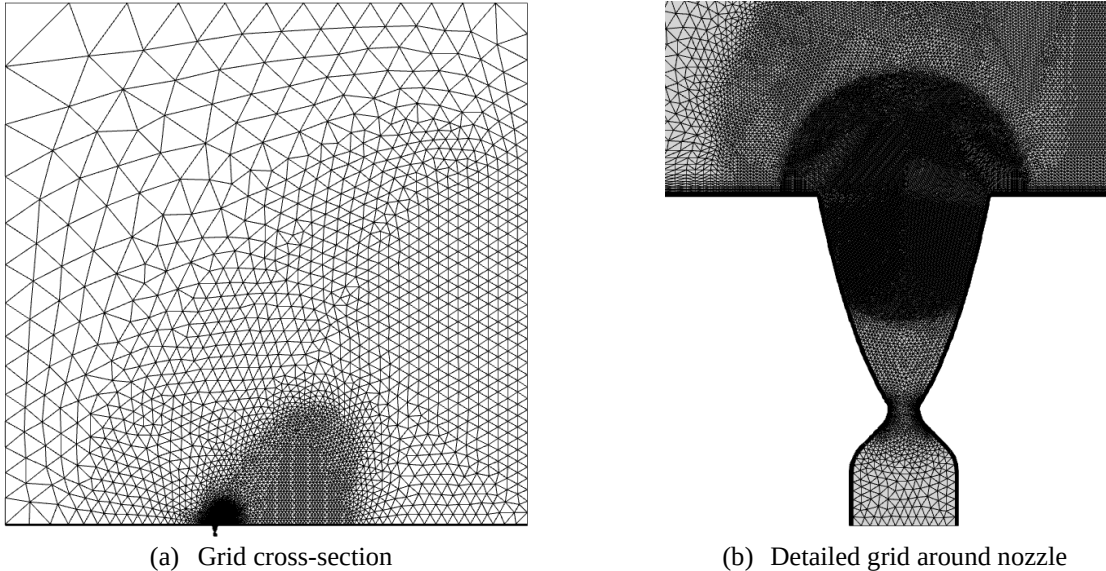


Fig. 1: Test Grid

Using the grid presented, a viscous boundary condition was set along the walls of the nozzle and the flat plate. The RCS inlet was set as a subsonic inflow boundary condition, and the rest of the bounding surfaces were set to a farfield boundary condition. As discussed above, the subsonic inflow boundary condition that models the plenum is controlled by two parameters: the total pressure ratio and total temperature ratio. The plenum pressure and temperature were defined as 1.5MPa and 1450K, respectively based off of literature review[17–19]. Martian atmospheric temperature,

pressure, and density at an altitude of 10km are used to define the freestream properties[20]. This corresponds to a pressure ratio of 270 and temperature ratio of 6.5 as defined by eq. (1) and (2). The simulation is run in air with a freestream Mach number of 3. The nozzle exit diameter of 0.07m was used as the reference length for the Reynolds number calculation and resulted in a Reynolds number of approximately $3e4$. The maximum allowable time step was calculated based off the Courant–Friedrichs–Lewy condition to be 0.007. Further refinement found that a timestep of 0.0007 was necessary to adequately converge the residuals. A roe flux construction scheme was used along with a van Albada limiter, which is augmented with a heuristic pressure limiter. For this case a ramping function duration of 20ms was implemented based off of a combination of literature review and an assessment of computing resources available[13,17].

V. Test case results

The problem methodology described above is implemented below for the test case. First, the on state is modeled to demonstrate the stability of the flow and appropriateness of the CFD settings. Then the off state establishment is described. Finally, the implementation of each ramp function is shown, along with the results from each of the simulations. The effectiveness of the ramping function is compared on the basis of numerical and flow stability.

A. Nozzle On Simulation

To simulate the RCS nozzle in an on state the pressure ratio of 270 and temperature ratio of 6.5 established previously from the literature are applied to the subsonic inlet boundary condition for the RCS jet. A time accurate simulation was run over 3000 steps using the flow solver settings discussed previously with the flow from this simulation shown in Fig. 2(a). As shown by the plots in Fig. 2(b), the residuals are well behaved and rise and fall as expected with little variation in the maxima and minima for each step. Comparing the flow shown in Fig. 2(a), one can see that the simulation reproduces many of the flow phenomena that are expected from a jet in a supersonic crossflow as detailed by literature sources [14,21]. This includes the recirculation zone slightly upstream of the nozzle as well as the formation of a barrel shock and large-scale structures downwind of the nozzle exit plane. A graphic of the expected flow phenomena for such a scenario is presented in Fig. 3. Given the stability of the simulation and success in reproducing the expected flow phenomena, the settings were deemed sufficient to move on to establishing the off state simulation.

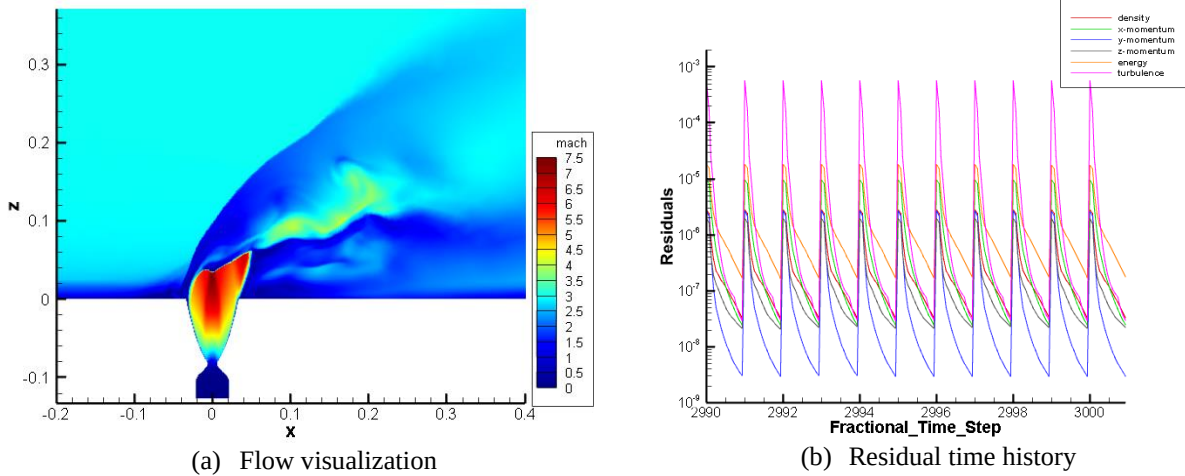


Fig. 2: Nozzle “on” simulation results.

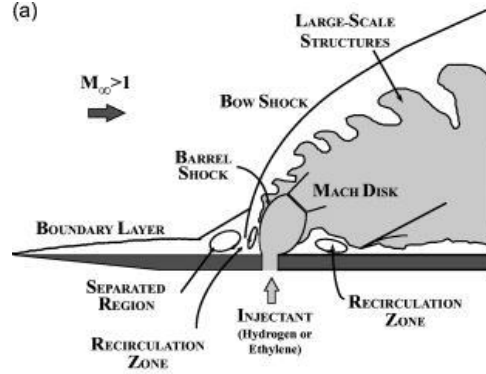


Fig. 3: Visualization of the jet interactions with a supersonic crossflow [22]

B. Nozzle Off Simulation

To establish the off state, the simulation was first run with an inviscid wall boundary condition at the jet plenum to demonstrate the ideal off condition. Then, samples were taken for the boundary condition in three different locations identified in Fig. 4. Equations (1) and (2) were then used to calculate the off-state boundary condition.

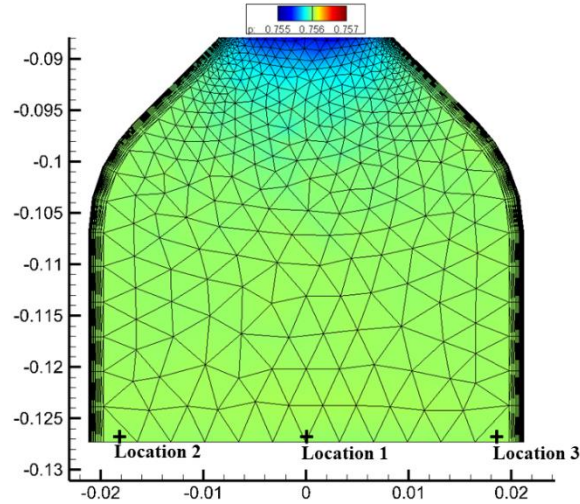


Fig. 4: Off Boundary Condition Sampling Locations

This sampling methodology resulted in a pressure and temperature ratio of 1.05 and 2.34 for the off state. The pressure ratio is close to 1 as expected, given the simplified setup, and the temperature ratio reflects the heating that occurs in the boundary layer. The results from the sampling can be seen in Table 2.

Table 2: Test Case Pressure and Temperature Sampling Data

	Location 1	Location 2	Location 3	Free stream	Ratio
Pressure	0.756042	0.756051	0.756048	0.714286	1.05
Temperature	2.090883	2.478979	2.463639	1	2.34

The next step is to run the same simulation but change the inviscid wall boundary condition to the subsonic inlet boundary condition defined by the calculated pressure and temperature ratio seen in Table 2. This simulation was then run for the same amount of time as the inviscid boundary condition simulation. The flow inside the nozzle for the two simulations can be compared in Fig. 5 which has limited the Mach number to 1 to enable proper comparison of the flow phenomena in the nozzle. As can be seen in Fig. 5, the results from the two simulations are similar but show some slight differences as expected due to sampling error which is the result of samples being taken very close

to the boundary but not at precisely at it. The pressure and temperature ratio used were determined to be acceptable for moving on to the next step of applying various ramping functions to bridge the off state to the on state.

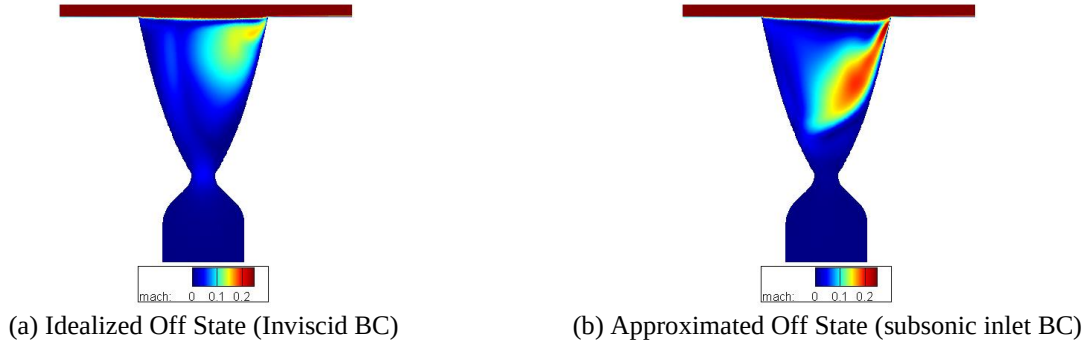


Fig. 5: Nozzle flow for idealized off versus off state mimicked by the subsonic inlet boundary condition

C. Nozzle Actuation Simulation

As discussed in the methodology section, three different ramp functions have been implemented: a step function, a linear function, and an exponential function. In this section the results of implementing each of these is discussed on the basis of commanded boundary condition values versus response, flow stability, and visual flow quality.

1. Step

The first ramp function implemented was the step function which follows eq. (3) above and defines PR_{off} and PR_{on} as 1.05 and 270 as is consistent with the above calculations. This information was coded into the FUN3D Python API, and the simulation was run for 10ms in the off state and 6ms in the on state. Additionally, the simulation was set up to output the pressure ratio and temperature of the boundary. This was done to allow for tracking of how well FUN3D was able to implement the commanded changes to the pressure and temperature ratios. Fig. 6 shows the commanded versus actual pressure during the simulation. One can see that the simulation is ultimately able to achieve the commanded pressure ratio but shows some transient behavior when the step is initially commanded.

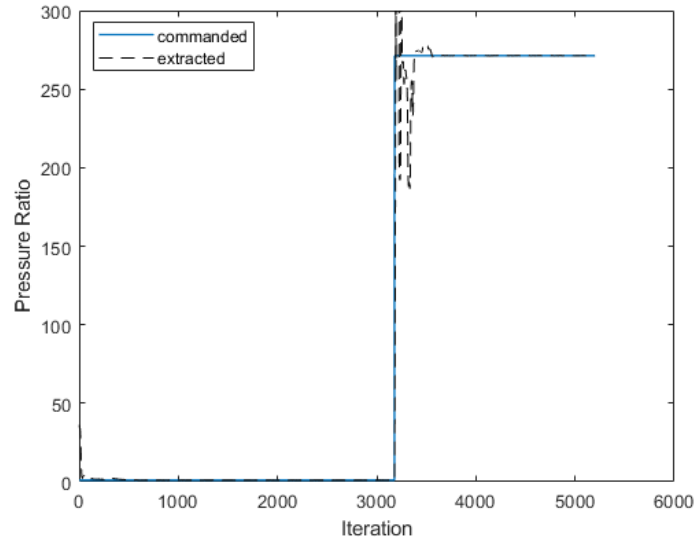


Fig. 6: Commanded vs. actual pressure change for step function

Additionally, a few slices are presented from key points in the simulation in Fig. 7. The first slice is from the off state and shows that the jet is indeed off. This slice was taken when the simulation converged on that state. The second slice is taken just after the step command was issued. As one can see, there is some nonphysical behavior in the chamber. This behavior is not accompanied by divergence in temperature or pressure solutions. The initial instability

was resolved quickly, and the flow was able to reach the on state pictured where the expected flow phenomena are achieved. Given the instabilities seen in the step response and the nonphysical flow behavior shortly after the step is implemented, it is clear that while the step function is acceptable in that it achieves both the on and off state without flow failure, the simulation would benefit from a gentler transition into the on state.

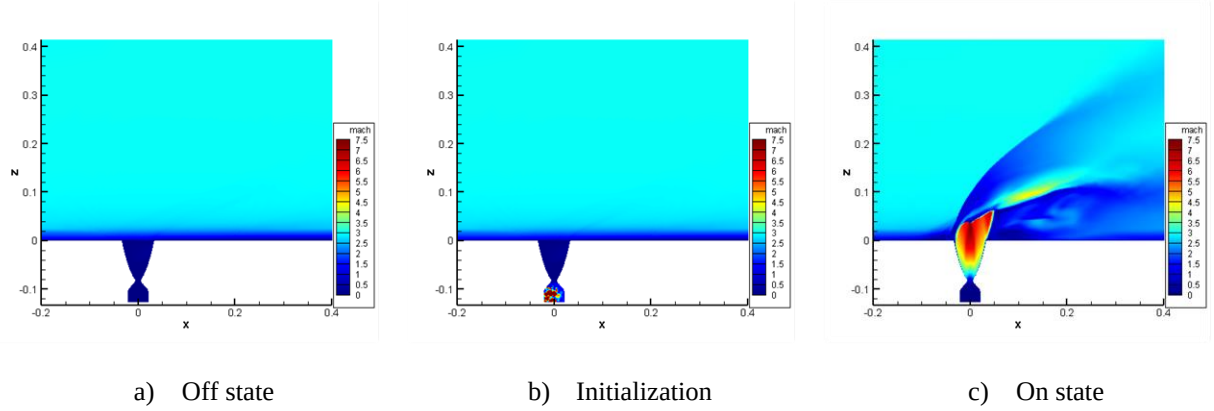


Fig. 7: Step function flow results

2. Linear

The next ramp function implemented was the linear ramp function which follows eq. (4). The ramp function is executed over 20ms as explained in the test case section. For this simulation 20ms comes out to 6360 steps. The off state is executed for 11ms or 3810 steps and the on state is executed for 13ms or 4135 steps. This is coded into the python API, and the simulation is run with the settings determined to be sufficient by the off and on state simulations. Fig. 8 shows the resulting commanded pressure change versus the extracted pressure change. The first thing to note on this comparison is that the extracted ramp function and the commanded ramp function are nearly identical. This simulation shows a significant improvement in the response over what was seen with the step response. There is very little deviation at the beginning of the ramping function, and the ramp continues to successfully follow the commanded pressure ratios throughout the simulation.

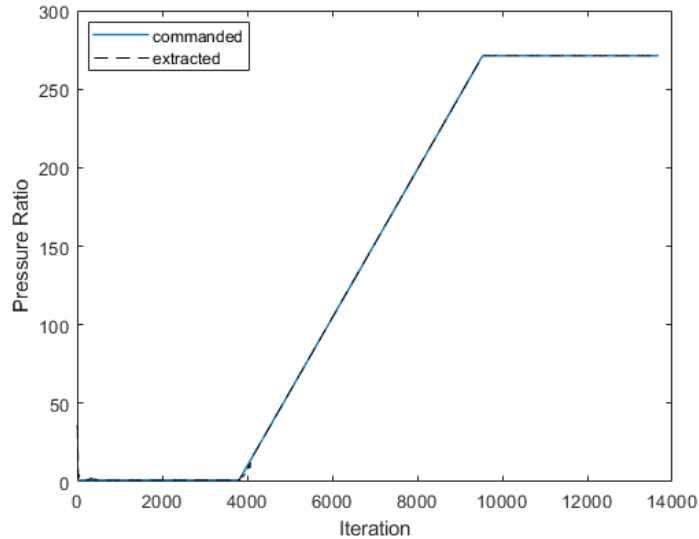


Fig. 8: Commanded vs. actual pressure change for linear ramp function

In Fig. 9 slices of the off state, initialization, and on state are presented. Once again, the off state is taken once the flow has fully initialized, the initialization slices are taken just after the ramp function starts in the jet initialization part of the run, and the on state is taken near the end of the run once the ramp function is complete and the jet has achieved the final pressure and temperature ratios. In the off and on state, the flow looks as expected and achieves

similar flow structures to what was seen in previous simulations. The initialization slice shows the flow beginning to exit the throat of the nozzle. This initialization shows significantly improved stability in the flow over what was seen in the step function. The nonphysical flow behavior at the bottom of the chamber seen with the step function has disappeared. The overall flow of the simulation shows good stability and demonstrates that simulation settings such as the flux limiter, the flux construction, the time step, the turbulence model, and the CFL can be selected to produce stable and accurate flow throughout the simulation. This is an important finding given that the flow phenomena are so different throughout the course of the simulation, lending credence to the idea that the optimal setting for each phase of the simulation may be different. This demonstrates that settings tuned to the on state can be used to achieve satisfactory results throughout the simulation.

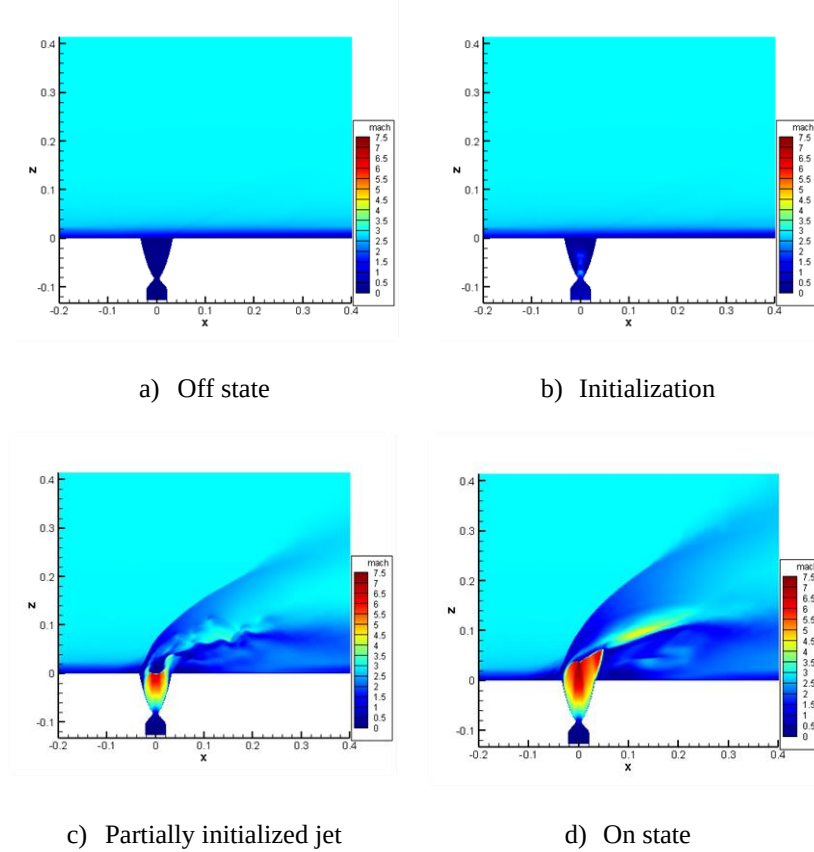


Fig. 9: Linear function flow results

3. Exponential

The success of the linear ramping function in resolving instabilities in the step function's flow led to the implementation of the final ramp function, an exponential function. Once again, this ramping function is completed with an 11ms off state, 20ms ramp phase, and a 13ms on state. Fig. 10 shows the commanded pressure versus the reported pressure. Overall, there is good agreement between the two in that they generally are following the same shape and there are no significant deviations from the commanded pressure ratio. However, there is some noticeable deviation at the beginning of the ramp function where the pressure ratio increases at a slightly decreased rate in comparison to what was commanded. The increased deviation here is somewhat expected in that the rate of increase in the pressure ratio at the beginning of the ramping function is over twice what is seen in the linear function but lower than what is seen in the step function. This leads to the conclusion that there is likely a limit to the rate of increase of the pressure ratio in order to maintain an appropriately responsive ramping function.

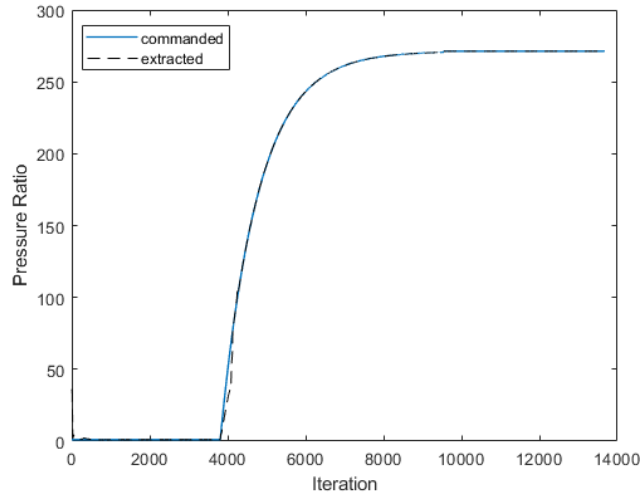


Fig. 10: Commanded vs. actual pressure change for exponential ramp function

Fig. 11 shows the off state, initialization, and on state slices for this simulation. The off state and on state once again look as expected and show no significant variance from previous simulations. The initialization slice shows the flow just beginning to exit the nozzle again and is comparable to the linear ramping function in visual behavior. Overall, the flow showed good stability and lacked numerical divergence or significant nonphysical behavior. This demonstrated that while the pressure ratio at the boundary showed some issues with decreased responsiveness, these issues did not translate to issues in the flow as it did in the step function. Thus, the slight deviation in the command versus actual response of the simulation is of little importance to the overall results of the simulation. This simulation demonstrated that it is feasible to implement an exponential ramping function for the pressure ratio and achieve good flow behavior throughout the simulation. Additionally, while there is some variance in the command versus actual response of the pressure ratio at the boundary, the lack of reflection of that issue in the flow behavior shows that steeper ramps may be achievable.

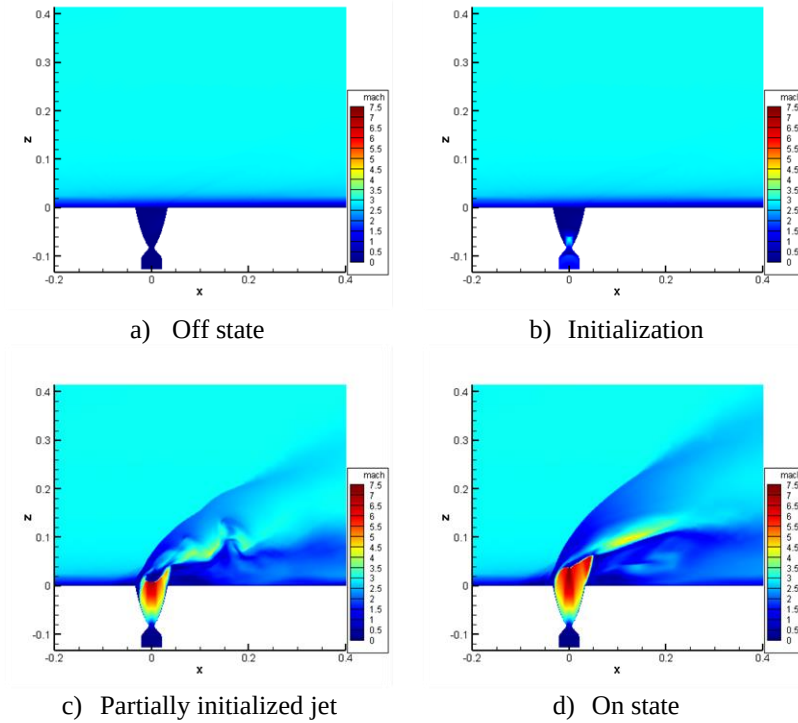


Fig. 11: Exponential function flow results

VI. Preliminary MSL methodology implementation

The completion of the jet in crossflow test case set the groundwork to implement the same methodology on the MSL vehicle. In this section a preliminary simulation of a linear ramp is shown to demonstrate the methodology on an EDL vehicle. This simulation was run using a 35 million node mixed element grid provided by NASA Langley based off of MSL flight geometry. The grid in Fig. 12 has significant density around the jets that quickly coarsens moving further into the wake.

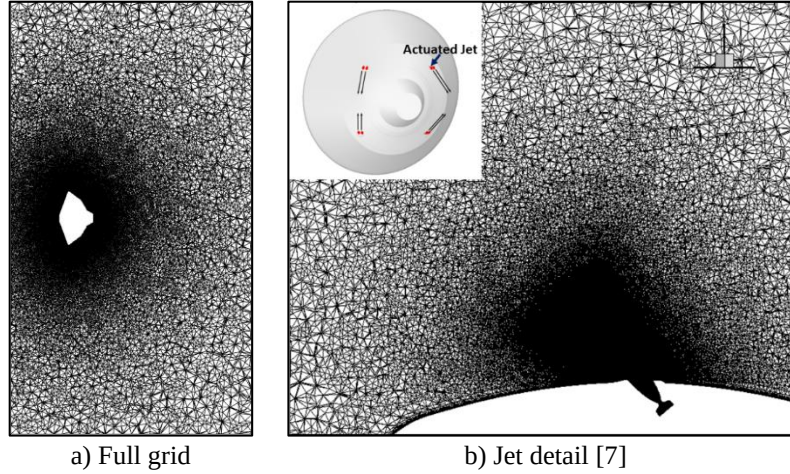


Fig. 12: MSL Grid

The MSL simulations are started with a static simulation. This approach was found to create better flow stability in comparison to starting the simulation as a time accurate simulation. As with the test case, the first step is to establish the on state and verify the selection of the CFD settings. For this simulation the atmospheric settings were based off of the Karlgaard et al. trajectory reconstruction at 692 seconds[23]. This resulted in a Mach number of 3.96, a temperature of 201K and a Reynolds number of 1.53×10^6 . The simulation was run at a 0 degree angle of attack. The pressure and temperature ratio were selected based off a chamber pressure of 1 MPa and temperature of 1450K leading to a pressure and temperature ratio of 7331 and 6, respectively[7]. The simulation was run with a dissipative LDFSS flux construction scheme coupled with a van Albada flux limiter. A time accurate simulation was used to verify these settings. A slice of the flow can be seen in Fig. 13. As can be seen from the slice, the flow shows an anomaly to the left-hand side of the opening of the nozzle, and the grid density is not sufficient to capture the jet wake interaction on the upper left area of the plume. The grid density was determined to be sufficient for the purposes of the demonstration case, and the flow anomaly is addressed in future simulations shown below. Overall, the simulation was considered a success in that the simulation showed good convergence behavior and expected flow behavior around the vehicle with the expected impact of the nozzle.

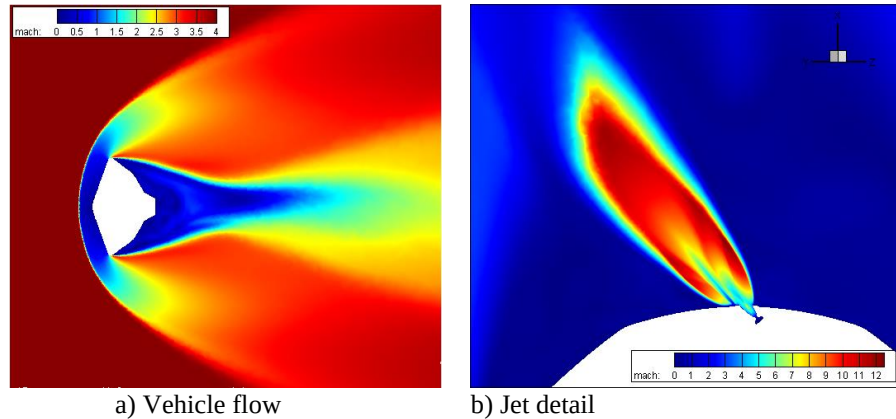


Fig. 13: On state verification

Next, the ideal off scenario with the inviscid wall boundary condition was run. Using the same sampling method demonstrated on the test case, the pressure and temperature ratios were found to be 0.43 and 3.65, respectively, for the off state. Table 3 shows the data for this.

Table 3: MSL Pressure and Temperature Sampling Data

	Location 1	Location 2	Location 3	Free stream	Ratio
Pressure	0.30942	0.308878	0.308882	0.714286	0.43
Temperature	3.65646	3.652086	3.651612	1	3.65

This information was then used to complete another time accurate simulation to verify the accuracy of the off state. Fig. 14 shows these results. The flow in the nozzle between the two simulations is similar. It shows some minor differences as was seen with the test case but was deemed sufficient to move on to the preliminary demonstration of the ramp function on the MSL grid.

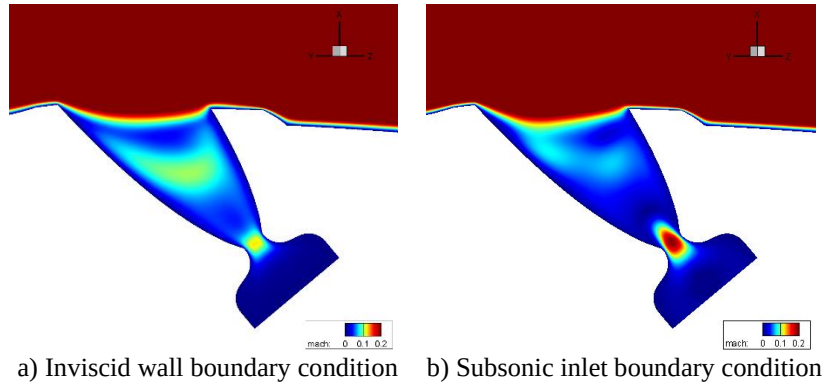


Fig. 14: MSL off state comparison

This set the stage for the implementation of a linear ramp function, which can be seen in Fig. 15 and was implemented based off eq. (4) for the pressure ratio. The temperature ratio was handled as a step as with all the previous simulations. The pressure ratio for this simulation had some difficulty achieving the commanded rate at the beginning of the simulation but quickly recovered. The off state is not shown on the ramp function but was run to convergence prior to the beginning of the initial actuation.

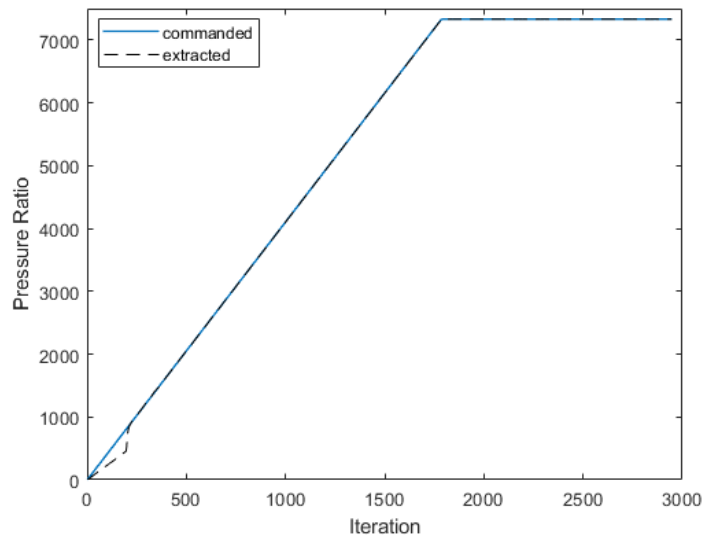


Fig. 15: Commanded vs. actual pressure ramp function for MSL demonstration

Additionally, the flow results around the RCS jet for the on case, which is analogous to a step function, and the linear ramp function, are presented in Fig. 16 for comparison. Both show similarly under-expanded flows for the jets. However, in the on state, there is a flow anomaly directly to the left of the nozzle as noted above in the verification of the simulation settings. This anomaly is resolved by the implementation of the ramp function. This points to the value in implementing a ramp function to be able to achieve more accurate results in difficult problems with complex flow interactions, such as the jet wake interactions seen on MSL.

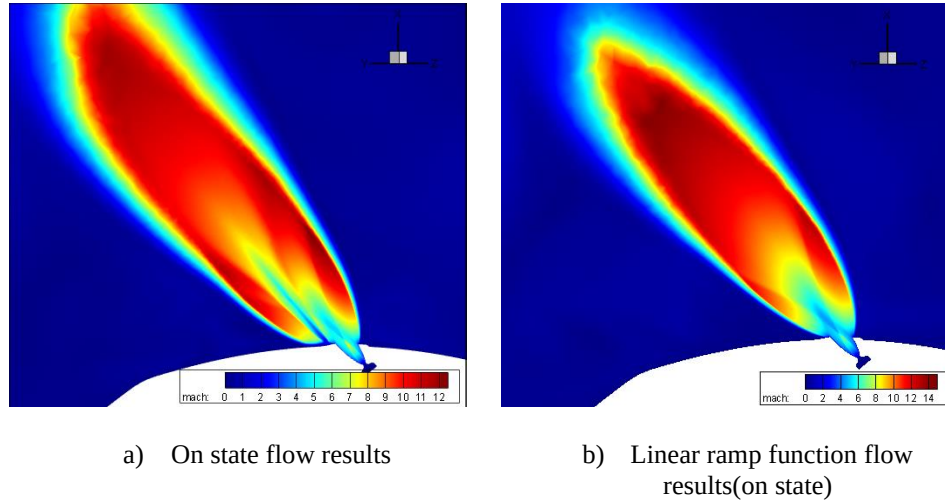


Fig. 16: Flow comparison for step versus linear ramp function for MSL

VII. Conclusion

This paper has established a methodology for the actuation of an RCS jet during a CFD simulation and presents a detailed demonstration on a test case of a jet in a supersonic crossflow. This demonstration showed a comparison of three different ramping functions: step, linear, and exponential. It demonstrated that CFD settings could be selected for these problems to achieve satisfactory flow resolution through the off state, initialization, and on states. The implementation of the exponential function demonstrated that steeper ramping functions could be achieved with satisfactory results, but there is a limit to the steepness of the ramping function before responsiveness issues are seen in the step function. The boundary of this limit was not identified here but could be the subject of future work. Additionally, a basic demonstration of this methodology is shown on the MSL vehicle. The MSL implementation of the methodology established some benefit of implementing a ramping function in the modeling of complex flow phenomena. The methodology set forth and demonstrated in this paper has set the groundwork for time accurate modeling of RCS jets in linked CFD, rigid body dynamics, and flight control systems simulations.

Acknowledgments

The team would like to thank the entry systems modeling project, Ashley Korzun, Li Wang, Kevin Jacobson, and Paul Tartabini for their guidance and support of this project. This project was also supported in part by the Partnership for an Advanced Computing Environment at the Georgia Institute of Technology, who provided computing resources and technical support.

References

- [1] Kazemba, C. D., Braun, R. D., Clark, I. G., and Schoenenberger, M. "Survey of Blunt Body Dynamic Stability in Supersonic Flow." *AIAA Atmospheric Flight Mechanics Conference 2012*, 2012. <https://doi.org/10.2514/6.2012-4509>.
- [2] Schoenenberger, M., Dyakonov, A., Buning, P., Scallion, W., and Van Norman, J. "Aerodynamic Challenges for the Mars Science Laboratory Entry, Descent and Landing." *41st AIAA Thermophysics Conference*, 2009. <https://doi.org/10.2514/6.2009-3914>.
- [3] Dyakonov, A. A., Schoenenberger, M., Scallion, W. I., Van Norman, J. W., Novak, L. A., and Tang, C. Y.

- “Aerodynamic Interference Due to MSL Reaction Control System.” *41st AIAA Thermophysics Conference*, No. June, 2009. <https://doi.org/10.2514/6.2009-3915>.
- [4] Johansen, C. T., Novak, L., Bathel, B. F., Ashcraft, S. W., and Danehy, P. M. “Mars Science Laboratory Reaction Control System Jet Computations with Visualization and Velocimetry.” <https://doi.org/10.2514/1.A32496>, Vol. 50, No. 6, 2013, pp. 1183–1195. <https://doi.org/10.2514/1.A32496>.
 - [5] Dyakonov, A. A., Glass, C. E., Desai, P. N., and Van Norman, J. W. “Analysis of Effectiveness of Phoenix Entry Reaction Control System.” *Journal of Spacecraft and Rockets*, 2011. <https://doi.org/10.2514/1.40965>.
 - [6] Inman, J. A., Danehy, P. M., Alderfer, D. W., Buck, G. M., and McCrea, A. “Planar Fluorescence Imaging and Three-Dimensional Reconstructions of Capsule Reaction-Control-System Jets.” <https://doi.org/10.2514/1.37116>, Vol. 47, No. 4, 2012, pp. 803–812. <https://doi.org/10.2514/1.37116>.
 - [7] Schoenenberger, M., Van Norman, J., Rhode, M., and Paulson, J. Characterization of Aerodynamic Interactions with the Mars Science Laboratory Reaction Control System Using Computation and Experiment. 2013.
 - [8] Edquist, K. T., Desai, P. N., and Schoenenberger, M. “Aerodynamics for Mars Phoenix Entry Capsule.” *Journal of Spacecraft and Rockets*, 2011. <https://doi.org/10.2514/1.46219>.
 - [9] Brock, J. M., Stern, E. C., and Wilder, M. C. “Computational Fluid Dynamics Simulations of Supersonic Inflatable Aerodynamic Decelerator Ballistic Range Tests.” *Journal of Spacecraft and Rockets*, Vol. 56, No. 2, 2019, pp. 526–535. <https://doi.org/10.2514/1.a34208>.
 - [10] Ernst, Z. J., Hickey, A., Robertson, B., and Mavris, Dimitri. “Impact of Roll Rate on Free-Flight CFD Modeling of Entry Vehicles.” *Journal of Spacecraft and Rockets (submitted for publication)*, 2021.
 - [11] Sahu, J., Fresconi, F., and Heavey, K. R. “Unsteady Aerodynamic Simulations of a Finned Projectile at a Supersonic Speed with Jet Interaction.” *32nd AIAA Applied Aerodynamics Conference*, No. June, 2014, pp. 1–35. <https://doi.org/10.2514/6.2014-3024>.
 - [12] Kim, J. S., Jung, H., Bae, S. H., Bae, D. S., and Kim, J. H. “Performance Evaluation of a 70 N Hydrazine Thruster According to the Variation of Characteristic Length.” *51st AIAA/SAE/ASEE Joint Propulsion Conference*, 2015. <https://doi.org/10.2514/6.2015-4162>.
 - [13] Dawson, M., Brewster, G., Conrad, C., Kilwine, M., Chenevert, B., and Morgan, O. Monopropellant Hydrazine 700 Lbf Throttling Terminal Descent Engine for Mars Science Laboratory. 2007.
 - [14] Grandhi, R. K., and Roy, A. “Effectiveness of a Reaction Control System Jet in a Supersonic Crossflow.” *Journal of Spacecraft and Rockets*, Vol. 54, No. 4, 2017, pp. 883–891. <https://doi.org/10.2514/1.A33770>.
 - [15] Arianegroup. Chemical Monopropellant Thruster Family. *Ariane Group*. 0. Volume 49. <https://www.space-propulsion.com/brochures/hydrazine-thrusters/hydrazine-thrusters.pdf>. Accessed Sep. 4, 2021.
 - [16] Rao, G. V. R. “Exhaust Nozzle Contour for Optimum Thrust.” *Journal of Jet Propulsion*, Vol. 28, No. 6, 1958, pp. 377–382. <https://doi.org/10.2514/8.7324>.
 - [17] Kim, J. S., Jung, H., Bae, S. H., Bae, D. S., and Kim, J. H. “Performance Evaluation of a 70 N Hydrazine Thruster According to the Variation of Characteristic Length.” *51st AIAA/SAE/ASEE Joint Propulsion Conference*, 2015. <https://doi.org/10.2514/6.2015-4162>.
 - [18] Kesten, A. “Analytical Study of Catalytic Reactors for Hydrazine Decomposition.” *NASA United Aircraft Research Laboratories G-910461*, 1968, p. 30.
 - [19] Seitzman, J. M. Thruster Pulsing. 1–3.
 - [20] Mars Atmosphere Model - Metric Units. <https://www.grc.nasa.gov/www/k-12/airplane/atmosmrm.html>. Accessed Oct. 11, 2021.
 - [21] Ben-Yakar, A. “Experimental Investigation of Mixing and Ignition of Transverse Jets in Supersonic Crossflows.” *Mechanical Engineering*, Vol. Doctor of, 2000, p. 217.
 - [22] Ben-Yakar, A., Mungal, M. G., and Hanson, R. K. “Time Evolution and Mixing Characteristics of Hydrogen and Ethylene Transverse Jets in Supersonic Crossflows.” *Physics of Fluids*, Vol. 18, No. 2, 2006, p. 026101. <https://doi.org/10.1063/1.2139684>.
 - [23] Karlgaard, C. D., Kutty, P., Schoenenberger, M., Shidner, J., and Munk, M. “Mars Entry Atmospheric Data System Trajectory Reconstruction Algorithms and Flight Results AIAA 2013-0028.” <https://doi.org/10.2514/6.2013-28>.