Development of Aerodynamic Loads Databases for the Space Launch System Booster Separation Event

Michael W. Lee[*],
NASA Langley Research Center, Hampton, VA

Derek J. Dalle[†],
NASA Ames Research Center, Moffett Field, CA

Michael M. Sanders[‡] and Carole J. Addona[§]
JSEG/Jacobs, Huntsville, AL

Booster separation is a mission-critical event within the orbital ascent of the Space Launch System (SLS). The complexity of the engine plume-affected, multibody, supersonic aerodynamics is compounded by the large span of the likely trajectory space. Characterization of the multiple input, multiple output system requires a combination of wind tunnel testing and computational simulation, but additional data processing is also required before the sparse, high-fidelity data can be fused into a continuous database with acceptable uncertainty quantification. This paper outlines the state of this approach as it has been applied to the most recent SLS booster separation aerodynamic loads database: that of the Artemis II launch vehicle.

Nomenclature

C_A	=	axial force coefficient
C_N	=	normal force coefficient
C_Y	=	sideforce coefficient
$\mathcal{D}$	=	nonlinear operator representing a database
d	=	distance between two points in a data space
M	=	Mach number
$\vec{p}$	=	parameter state vector
$\vec{q}$	=	quantity of interest vector
q	=	dynamic pressure
R	=	IDW interpolation hypersphere radius
T_{BSM}	=	booster separation motor thrust magnitude
T_{CSE}	=	core stage engine thrust magnitude
T_{SRB}	=	booster thrust magnitude
U_{C2C}	=	code-to-code uncertainty
U_{C2T}	=	code-to-tunnel uncertainty
UC_Y	=	sideforce coefficient uncertainty
U_{mod}	=	modeling uncertainty
U_{tot}	=	total uncertainty
w	=	inverse-distance weights
α	=	body-centered angle of attack
β	=	body-centered angle of sideslip
ΔeC_n^C	=	difference in interpolation errors for the core stage yawing moment
ΔeC_N^R	=	difference in interpolation errors for the right booster normal force
Δx	=	preseparation core-relative booster nose distance in the core stage x-axis

[*] Research Aerospace Engineer, Configuration Aerodynamics Branch, NASA Langley Research Center, AIAA Member.

[†] Research Aerospace Engineer, Computational Aerosciences Branch, NASA Ames Research Center

[‡] Modeling and Simulation Engineer, EV42 Guidance, Navigation, and Mission Analysis Branch, JSEG/Jacobs

[§] Liftoff/Separation Technical Lead, EV42 Guidance, Navigation, and Mission Analysis Branch, JSEG/Jacobs

Δy	=	preseparation core-relative booster nose distance in the core stage y-axis
Δz	=	preseparation core-relative booster nose distance in the core stage z-axis
$\Delta\theta$	=	preseparation core-relative booster pitch angle in the core stage axis
$\Delta\psi$	=	preseparation core-relative booster yaw angle in the core stage axis
$\Delta\phi$	=	preseparation core-relative booster roll angle in the core stage axis
ϵ	=	pseudorandom trajectory dispersion parameter
σ	=	standard deviation of a normal distribution
BSM	=	Booster Separation Motor
CAPE	=	Computational Aerosciences Productivity and Execution
CFD	=	Computational Fluid Dynamics
CFL	=	Courant-Friedrichs-Lewy condition
F&M	=	Force(s) and Moment(s)
IDW	=	Inverse-Distance Weighting
LAS	=	Launch Abort System
LOX	=	Liquid OXygen
MPCV	=	Multi-Purpose Crew Vehicle
RANS	=	Reynolds-averaged Navier-Stokes
SLS	=	Space Launch System
UPWT	=	NASA Langley Unitary Plan Wind Tunnel

I. Background and Motivation

Booster separation has been a component of NASA rocket ascent trajectories in many of its launch vehicles, including the space shuttle and the Titan rockets. This event remains a staple of the Space Launch System (SLS) mission profiles. The liquid-propellant SLS core stage, powering four RS-25 engines, augments the much more powerful solid-propellant boosters during liftoff and initial ascent, which are mounted to either side of the vehicle. The boosters provide the thrust necessary to lift the rocket, its fuel, and its payload off the launch pad and through the lower atmosphere; at liftoff, the boosters provide approximately 75% of the total vehicle thrust. Once the boosters begin to run out of fuel, which happens after the rocket sustains maximum dynamic pressure but before the core stage itself jettisons from the interim cryogenic propulsion stage, the boosters separate and fall earthward as the core stage continues its ascent. This event is powered by several booster separation motors set between the boosters and the core stage. These motors push the boosters away from the core stage, although there is no active thrust control during the firing process.

During booster separation, as depicted in Figure 1, three separate bodies traveling at supersonic speeds are moving very close to one another; after initial separation, any core stage recontacts threaten the success of the mission, and possibly the safety of the crew. This risk is most notable in the first several feet of separation, where the boosters must clear the attachment hardware that had until that point ensured the three bodies remained locked together. Further downstream, there is also a less critical risk of the boosters contacting one another. While this eventuality is not desirable, it has not been deemed mission-critical because the boosters will not be recovered. The likelihood of such contact events is calculated with Monte Carlo booster separation simulations, in which many permutations of operating conditions and system predictions inform a separation trajectory envelope that enables a safe and successful mission.

The induced aerodynamics of this separation event are an essential element in predicting the separation trajectory envelope. The vehicle aerodynamics in this flight window are inherently complex: the core stage creates a large bow shock, with which the boosters may interact during separation; the RS-25 plumes are underexpanded due to the reduced atmospheric density; the dwindling booster engine plumes also complicate the vehicle wake; the booster separation motor plumes impinge directly on the core stage during initial separation; and the booster positions relative to the core stage can vary within the resulting flowfield. Core recontacts are most likely in the first several feet of separation, in which region viscous effects play a more significant role.

In this paper, we outline the latest procedure for developing an induced aerodynamic loads database for the SLS booster separation event. Such an effort includes wind tunnel testing, thousands of computational fluid dynamics (CFD) simulations, and a procedure for interpolating the resulting data space in such a way that is as high-fidelity as possible. The development process is organized into two sections: that of the nominal database (§II) and that of the uncertainty model (§III). Trajectory simulations are then summarized (§IV) to illustrate the influence of the resulting aerodynamic database on the Artemis II booster separation event.

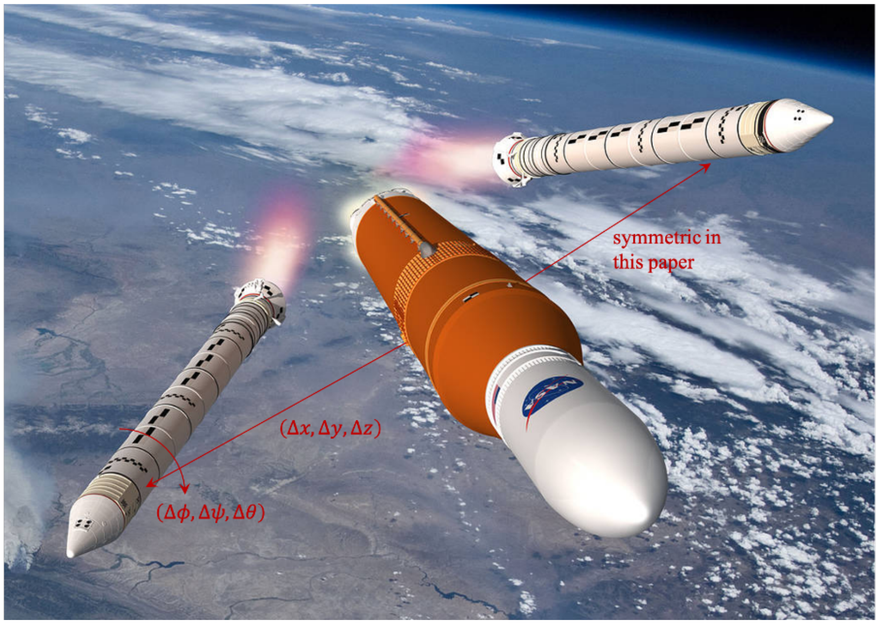

symmetric in this paper
$(\Delta x, \Delta y, \Delta z)$
$(\Delta\phi, \Delta\psi, \Delta\theta)$

Fig. 1 Illustration of the SLS booster separation event, with core stage-relative booster position and rotation parameters labeled.

II. Nominal Database Development

As with other SLS aerodynamics databases, the booster separation database enables a dispersion of force and moment (F&M) coefficients with bounds defined by the uncertainty model and nominal values at the bound centers. Within a single Monte Carlo trajectory simulation, several uncorrelated dispersion parameters ϵ, bounded between -1 and 1, are held constant. During trajectory simulation, these dispersion parameters alter the nominal database loads within the uncertainty bounds. For example, for the right booster sideforce, the dispersed nominal value would be found based on the following equation:

$$C_{Y,disp}^{RSRB} = C_{Y,nom}^{RSRB} + \epsilon_{Y}^{RSRB} U C_{Y}^{RSRB} \tag{1}$$

In this section, the development and prescribed treatment of the nominal database values, e.g., $C_{Y,nom}^{RSRB}$, is outlined.

There are three data sources that supply the booster separation database: FUN3D simulations, which serve as the nominal source; a test in the NASA Langley Unitary Plan Wind Tunnel (UPWT), which provides data for uncertainty quantification; and OVERFLOW simulations, which also provide data for uncertainty quantification. As illustrated by the data pipeline in Figure 2, the FUN3D data comprise a table of force and moment coefficients, which is then interpolated by the data users.

A. FUN3D Run Matrix

The booster separation event can occur within a range of freestream flow conditions. After the initial separation, each booster is pushed away from the core stage by eight separate motors. Although the first several feet of booster travel are where core stage recontacts are most possible, the booster separation must be tracked for several hundred feet downstream in order to ensure that separation anomalies will not affect the rocket mission. During Artemis I booster separation database development [1, 2], a set of thirteen freestream and positional parameters were identified by which the booster separation data space was defined:

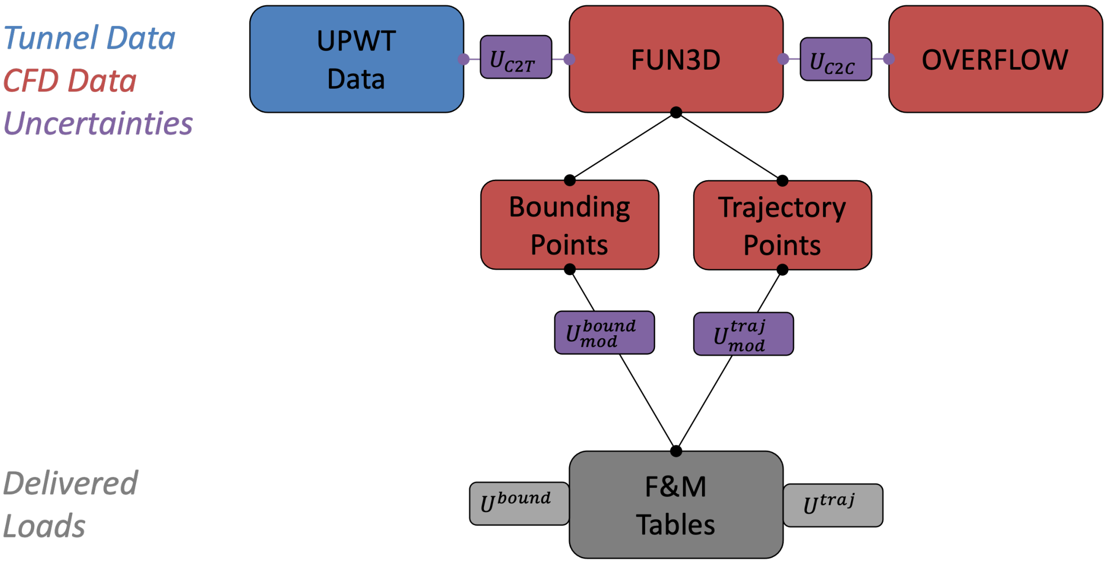


Fig. 2 Data pipeline for the Artemis II SLS booster separation database.

1,2) α, β: the core stage aerodynamic incidence and sideslip angles;
3,4) q, M: the freestream dynamic pressure and Mach number;
5) T_{CSE}: the average core stage engine (CSE) thrust;
6) T_{SRB}: the right solid rocket booster (SRB) thrust;
7-9) Δx, Δy, Δz: the right booster nose position, in the core body axis, relative to its separation position;
10-12) $\Delta\theta$, $\Delta\psi$, $\Delta\phi$: the right booster pitch, yaw, and roll, about the core body axis, relative to its preseparation orientation; and
13) T_{BSM}: the average right booster separation motor (BSM) thrust.

This same parameter set served as the baseline for the Artemis II database development. The left booster characteristics were set symmetrically, e.g., at any given run point, $\Delta x_{RSRB} = \Delta x_{LSRB}$, $\Delta\psi_{RSRB} = -\Delta\psi_{LSRB}$. This simulated symmetry stems from the assumption that the aerodynamics affecting one booster do not significantly influence the aerodynamics of the other booster; the resulting database can be queried independently for LSRB and RSRB values asymmetrically without violating this assumption of aerodynamic isolation. Verification and validation tests were conducted to lend credence to this assumption. The averaged thrusts were likewise verified as an acceptable simplification.

Two separate run matrices were organized to fill the data space as well as possible. The first run matrix spanned a wide range of possible conditions with a somewhat uniform point distribution. This, however, came at the cost of high point concentrations in any particular data space locale. After a preliminary database was released based on this encompassing run matrix, trajectory dispersions were simulated to identify a more focused data space of interest. A second run matrix was then organized based on the thirteen-dimensional bounds of these initial trajectory dispersions, this time with a less structured distribution that was based on a space-filling Sobol sequence. Plots in Figure 3 illustrate how the original CFD run matrix spanned a wide data space and then the Sobol sequence run matrix provided additional points where trajectories were more likely to travel. It is worth noting that these flattened data space visualizations do not fairly represent the sparsity of the data space; when a 13 convex hull was defined around the Sobol sequence points, all but approximately ten of the original run points were outside of that convex hull. This means that the initial trajectory simulations were based largely on interpolations between a few dozen, out of thousands, of CFD points, where with the addition of the Sobol sequence points, far more CFD data exist where the trajectories are expected to exist.

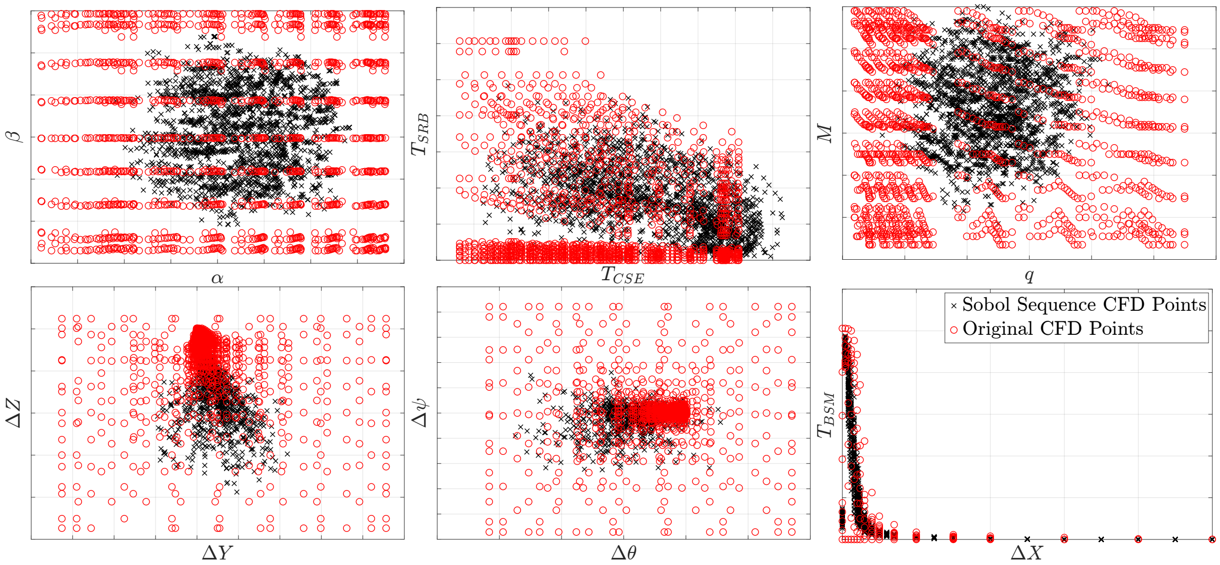


Fig. 3 CFD run points distributed in thirteen dimensions, plotted as flattened two-dimensional spaces.

B. FUN3D Simulations

Although the Space Shuttle Program booster separation aerodynamics database was based primarily in wind tunnel data, both technical and financial factors informed the decision to base the SLS booster separation aerodynamics databases primarily on computational data. Wind tunnel tests, as discussed in §III.B, were still used to develop the uncertainties that bound the nominal database, but the nominal data were gathered from computational sources. Namely, FUN3D was used to simulate thousands of points within the high-dimensional parameter space. Organization of so many simulations was made more manageable by the recently developed Computational Aerosciences Productivity and Execution (CAPE) software.

This work used FUN3D version 13.1 [3]. FUN3D is a fully unstructured Navier-Stokes flow solver developed at NASA Langley Research Center. The solutions are steady-state Reynolds-averaged Navier-Stokes (RANS) solutions with turbulence closure from the one-equation Spalart-Allmaras [4] turbulence model. The flow is considered turbulent everywhere. No-slip, cold walls with the ratio of wall temperature to reference temperature set to 1 were used to model the surface boundaries of the vehicle. The simulations used the perfect-gas version of FUN3D, so the exhaust is modeled as high-temperature air and cannot match all three conditions: thrust, mass flow rate, and temperature. The throat and subsonic regions of each nozzle were modified to match thrust and mass flow but not temperature. Thrust conditions in the flow-through faces of 22 different nozzle plenums (where simulated fuel and oxidizer enters the flow) were set using total to freestream ratios for pressure and temperature. This is FUN3D boundary condition 7011. When the booster separation motors (BSMs) are not firing, these boundaries are treated as no-slip walls.

Individual surface meshes representing the geometry of the SLS core and both boosters were prepared using the ANSA[5] preprocessor. Figure 4 shows detail of the resulting surface mesh. Here the boosters are shown in the stowed position prior to separation, and the colors represent the different computational surfaces defined during surface mesh generation.

Each surface along with mesh refinement sources were then translated and rotated to its correct position based on values in the CFD run matrix. The Cart3D[6] tool `intersect` was used to create a unique triangulation by combining the resulting meshes with farfield boundaries. Then, an initial volume mesh with approximately 12 million points was generated from the resulting triangulation using the AFLR3 [7] software. Each case in the CFD run matrix receives its own unique volume mesh. During runtime, two feature-based off-body (frozen viscous layer) mesh adaptation cycles were performed by FUN3D using Mach number as the feature scalar key and the `delta-l` option, with the final volume mesh containing approximately 25 million points. Figure 5 shows a slice of a representative volume mesh near the forebody of the core body before and after feature-based adaptions were applied. Booster separation occurs around Mach 4.0 allowing for relatively easy convergence when compared to a similar case run at transonic Mach numbers. The solver was run steady state and relatively aggressively, with a maximum CFL number of 200. Ideally, cases converged after 3000 iterations with the BSMs off and after 4000 iterations with the BSMs firing. Each case

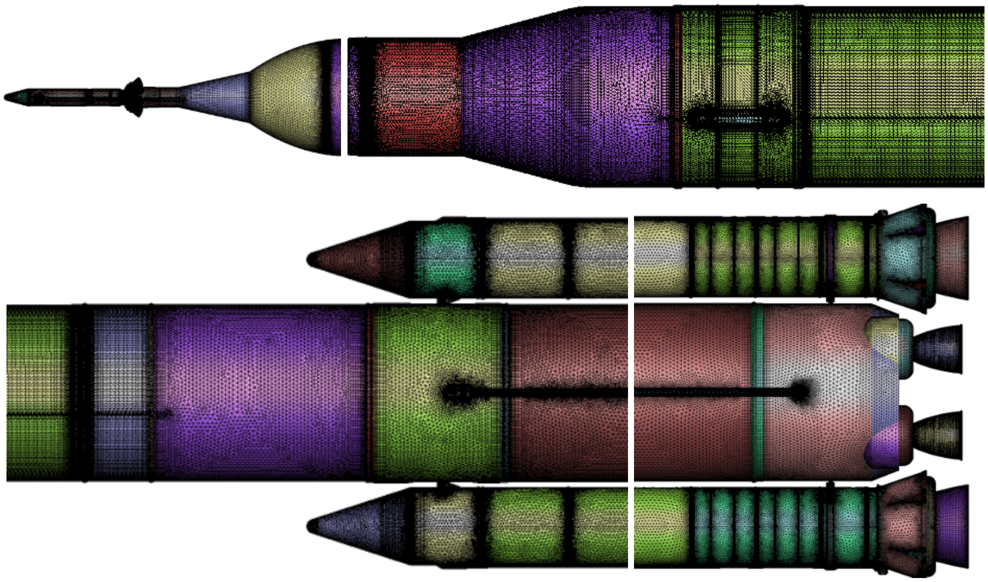
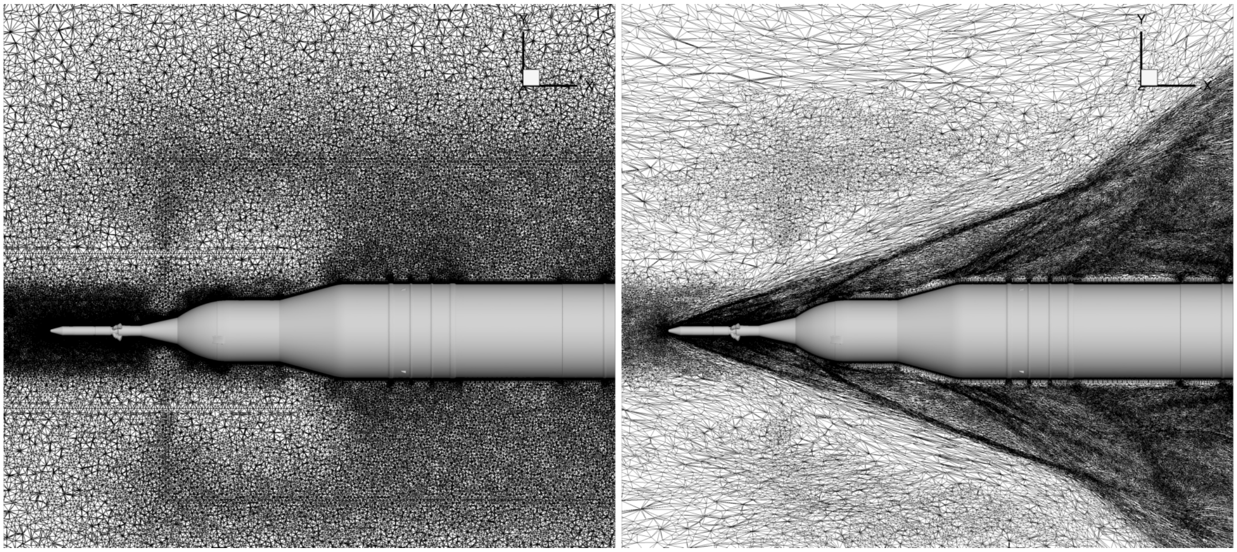


Fig. 4 Surface mesh details at Δx = 0 ft.

was post-processed and inspected individually for convergence, which was determined when bulk flow variables (C_A, C_N, etc.) were observed to be unchanging or to have reached a stable limit cycle. Typical residual drops spanned several orders of magnitude before solution convergence was identified. Typically, cases converged after 2000 or 3500 core hours (including mesh generation), depending on if the BSMs were off or on, respectively. All runtime case management and post-processing was performed using the CAPE [8] software package. CAPE provided a set of tools specifically designed to streamline database creation by automating problem setup and data extraction steps.



(a) Initial mesh

(b) Mesh after feature-based adaptions

Fig. 5 Representative $z = 0$ volume mesh slice.

C. Interpolation

Despite the large number of data points delivered via FUN3D simulations, in the context of the high data space dimension, the database is in fact relatively sparsely populated. As such, a significant majority of trajectory simulation samples reside amidst the delivered points, rather than precisely at the delivered points. Interpolation of the delivered data tables is thus mandated.

In lower-dimensional data spaces, due to the generally smooth nature of aerodynamic trends, the nature of this interpolation does not significantly affect the resulting values. However, in the high-dimensional, multibody domain of booster separation, the interpolation method is an important aspect of the database itself.

Interpolation via inverse-distance weighting (IDW) was selected due to its robust behavior in high dimensions, its capacity to handle sparsely populated data spaces, and its simplicity and efficiency of implementation. In particular, the modified Shepard's method was selected. The database itself can be represented by the nonlinear operator $\mathcal{D}$, which receives a parameter state vector $\vec{p}$ and returns a quantity vector $\vec{q}$, which in this case comprises force and moment coefficients on the three separating bodies.

$$\vec{q} = \mathcal{D}(\vec{p}) \tag{2}$$

Given the tabulated structure of the database and the IDW interpolation scheme, where the point of interest is not virtually identical to a provided point, the nonlinear operator $\mathcal{D}$ reduces to a weighted sum of N tabulated quantities $\vec{q}_{DB}$.

$$\mathcal{D} \equiv \left(\sum_{n=1}^{N} w_n \vec{q}_n\right) \Bigg/ \left(\sum_{n=1}^{N} w_n\right) \tag{3}$$

Via the modified Shepard's method, the weights are themselves a function of the queried parameter space point $\vec{p}_q$:

$$w_n \equiv \left(\frac{\max\left(0,\ R - d(\vec{p}_n,\ \vec{p}_q)\right)}{R d(\vec{p}_n,\ \vec{p}_q)}\right)^2 \tag{4}$$

where R is the radius of a hypersphere that negates the influence of nonproximal database points and $d(\vec{p}_n,\ \vec{p}_q)$ is the Euclidean distance between the nth database point and the query point. For proper point distance calculation, the data space was nondimensionalized such that all axes were unitless, e.g., axes with units of length and angle could be added together to compute d. Based on the recommendations outlined in Algorithm 661 of the NetLib TOMS library [9], the radius of influence R is set on a query-by-query basis such that no more than the closest 40 database points are used for any given interpolation.

Although the CFD run matrix was organized in thirteen dimensions, there is no guarantee that this full-dimensional space is a reasonable one for the tabulated data space. The data sparsity naturally increases exponentially as the dimension increases, but if the data space is itself not sensitive to one or more of these dimensions, then this increased sparsity reduces the database fidelity without any benefit of meaningful parameter dependence. To mitigate this, a dimension flattening study was performed to inform the final data space dimension: a subset of the initial thirteen.

The dimensional dropout analysis was organized as follows. A random subset of the FUN3D data was set aside as the test data set and the remaining points were treated as the interpolation data set. The latter basis was then used to interpolate the coefficients at the test points eleven times: once with all thirteen dimensions and once with each (except the three core-relative booster Euler angles) dimension flattened, viz., not considered in the IDW process but with all points still retained in the data set. The three Euler angles were never flattened because doing so would require the booster coefficients to be transformed into the core stage body axis, and the known importance of the angles to the data space meant that this effort was not justifiable. The root-mean-square of the difference between test point interpolation errors, e.g., $eC_Y \equiv |C_Y^{interp} - C_Y^{CFD}|$, between the unflattened and each of the flattened interpolated values was then recorded. This process was then performed fifty times with different random points set aside for testing; the random points were selected such that 1) each point was used for testing approximately the same number of times and 2) the test points were uniformly distributed in the Δx space because the database is organized based on that parameter. A larger difference between a flattened and unflattened interpolation set indicates that the flattened parameter affects the quantity of interest more, and that it is therefore less valid for flattening.

Representative results of this parameter flattening analysis are presented in Figure 6. Only two coefficient results are plotted: those of the RSRB normal force and the core stage yawing moment, which are two of the more important coefficients. Mean, maximum, and minimum sensitivities are plotted based on the 50 trials, for each flattened parameter as labeled on the x-axes. As with many sensitivity analyses, a notable difference in parameter importance should present itself as a roughly order-of-magnitude difference when comparing one dropped parameter sensitivity to another. For example, the core and booster thrusts are notably less sensitive parameters for both plotted coefficients. However, the booster normal force is more sensitive to freestream Mach number and dynamic pressure than is the core yawing

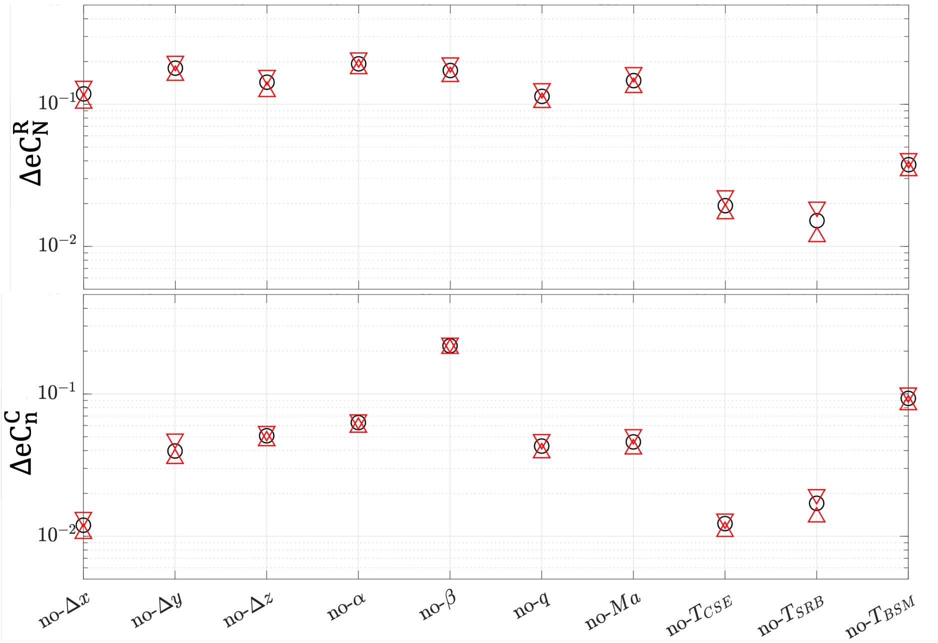


Fig. 6 Dimension flattening sensitivities (min, mean, and max from several trials) for the right booster normal force (top) and the core stage yawing moment (bottom).

moment; this makes sense intuitively, as the booster loads are more dependent on the nature of the engine plumes and bow shock structure because they can impinge on them during the separation event.

In order to avoid skewing the physics representation, the final database must account for all parameters that are important for at least one critical F&M coefficient. From this parameter sensitivity analysis, when considering all relevant coefficients and all flattened parameter sensitivities, it was determined that the parameters T_{CSE} and T_{SRB} were unimportant enough that they could be flattened in the delivered nominal database.

III. Uncertainty Model Development

The nominal database values determine the F&M trends within the parameter space, but the uncertainty model determines the magnitude of those trends for a particular dispersed trajectory simulation. The uncertainty model accounts for mutually independent uncertainty sources that comprise the epistemic and aleatoric differences between the database and reality. In total, the uncertainty model accounts for:

- differences between engine plume physics as captured by two different flow solvers;
- differences between simulated and physically observed supersonic flow physics of booster separation with only separation motor plumes; and
- differences between simulated physics and physics inferred via database interpolation.

These three uncertainty sources, in order, are hereby referred to as code-to-code uncertainty U_{C2C}, code-to-tunnel uncertainty U_{C2T}, and model uncertainty U_{mod}. Assuming statistical independence, these sources are reconciled into a single uncertainty model U_{tot} via a root-sum of squares.

$$U_{tot} = \sqrt{U_{C2C}^2 + U_{C2T}^2 + U_{mod}^2} \tag{5}$$

The uncertainty model is computed separately for each of the eighteen F&M coefficients. In the remainder of this section, the calculation of the three uncertainty sources will be outlined.

A. Code-to-code Uncertainty

A subset of the FUN3D cases were randomly selected to be simulated with the OVERFLOW flow solver [10] for code-to-code comparisons. OVERFLOW is a structured overset grid RANS flow solver, and version 2.3e was used for this analysis. Seven cases were randomly chosen at 16 different Δx values for a total of 112 cases.

The geometries of the FUN3D and OVERFLOW simulations were nearly exact. Two small differences existed: one at a strut in the aft attach hardware, and another slight difference at the SRB forward BSM nozzles. Both of these differences were considered negligible for overall vehicle forces and moments.

The OVERFLOW mesh had an order of magnitude more grid points than the FUN3D mesh. OVERFLOW simulated the nozzle plumes with a mixing (nonreacting) multispecies approach where each engine type (BSM, SRB, and CSE) had effective properties for their exhaust gas.

These OVERFLOW simulations used the HLLC flux differencing scheme, the SSOR implicit scheme, and the SA turbulence model [4]. A locally varying time-step size with constant CFL was employed. Simulations were started with full multigrid sequencing, and after the solution was somewhat developed, 25 cycles of adaptive mesh refinement (AMR) [11] were conducted for the off-body Cartesian grids. The simulation continued until the booster and core forces and moments reached a steady state. Each case simulated in this matrix had its own SRB position.

The SRBs were positioned using two methods, where the method choice depended on the distance of the SRBs and center body from each other. If the three bodies were far away from each other, the OVERFLOW built-in Geometry Manipulation Protocol (GMP) [12] was used to position the SRBs. If any of the three bodies were close to each other, a Python routine was used to position the SRBs and the Pegasus5 [13] code was used to preprocess the domain connectivity. Figure 7 shows six examples of the structured overset surface mesh for these SLS simulations. From left to right, top to bottom, the frames show the LAS abort nozzles, MPCV umbilical, SRB forward BSMs, LOX feed line forward fairing, SRB forward attach hardware, and SRB aft BSMs. The images give an idea of the overset topography of the mesh system as well as the refinement level.

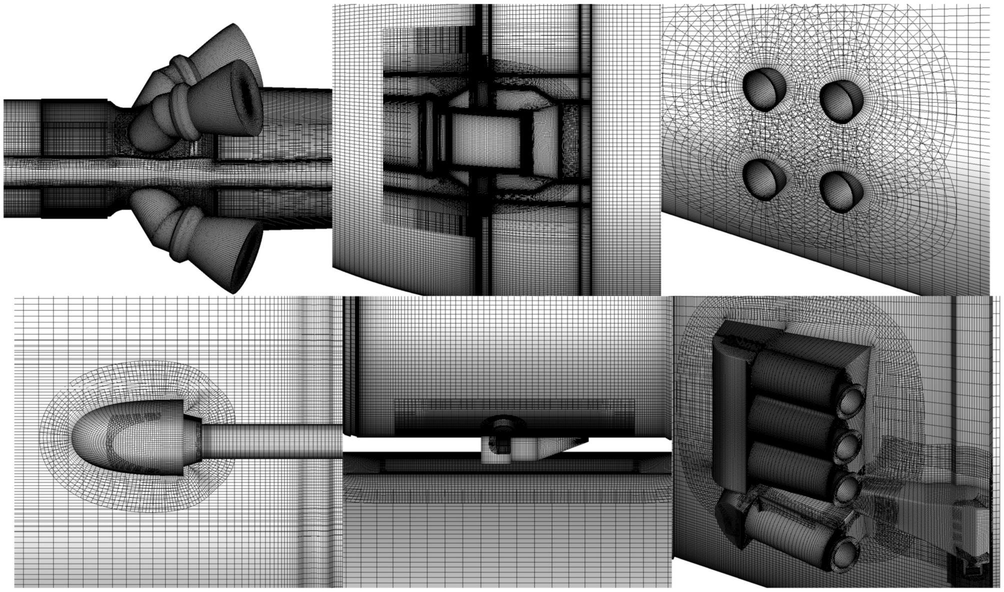


Fig. 7 Examples of the structured overset surface mesh used for OVERFLOW simulations.

Figure 8 shows a slice of the volume grid on the symmetry plane for the structured overset grid system. The center body and the left SRB are seen in the image. The curvilinear near-body grids of the center body can be seen, as well as the Cartesian off-body grids that fill the domain. The Cartesian off-body grids were automatically generated by OVERFLOW [14]. AMR in the off-body grids was employed and can be seen predominantly in the plume region of the center body.

Each grid had a different SRB position and AMR results, but the total number of grid points was limited to approximately 425 million. The surface grids and near-body volume grids were generated using Chimera Grid Tools and its Tcl scripting library [15, 16].

The domain connectivity for all OVERFLOW simulations utilized a hybrid approach between the Pegasus5 preprocessing code [13] and the Domain Connectivity Function (DCF) [17] process that is internal to OVERFLOW. The Pegasus5 code performs the overset connectivity, performs hole-cutting of any overlapping grids that are internal to one of the bodies, and it optimizes the overset regions based on matching cell size. The DCF approach is built into OVERFLOW, it does not optimize overset regions, and it uses a relatively coarse hole-cutting procedure that requires more

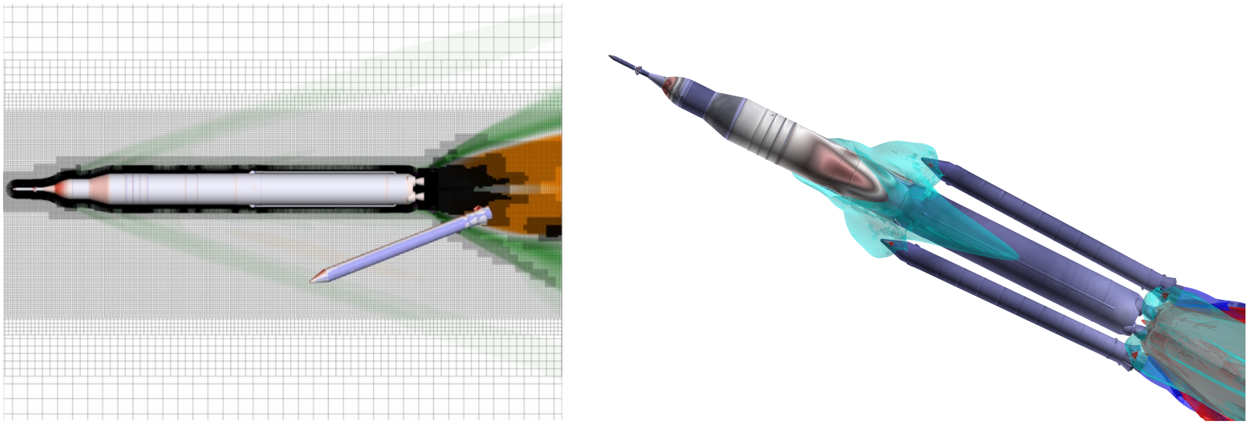


Fig. 8 Post-AMR example of the structured overset volume mesh used for OVERFLOW simulations (left) and a snapshot of the simulation itself (right).

user input and expertise. The CAPE software suite [8] was used for case and file management, solution post-processing, and the development of the databases required for the code-to-code uncertainty quantification.

An example of an OVERFLOW booster simulation is shown in Fig. 8. In the figure, the plume exhaust is visualized where the BSM exhaust is cyan, SRB exhaust is blue, and CSE exhaust is red. The surface is colored by pressure coefficient. For the case visualized, BSM plume impingement is significant.

Plots in Figure 9 present the absolute differences in C_Y^{RSRB} and C_N^{RSRB} between all repeated FUN3D and OVERFLOW simulations, as a function of Δx. The plots also present the 99.7th percentile as computed by grouping these points into several Δx bins. This upper percentile denotes the code-to-code uncertainty as it was used in the uncertainty buildup.

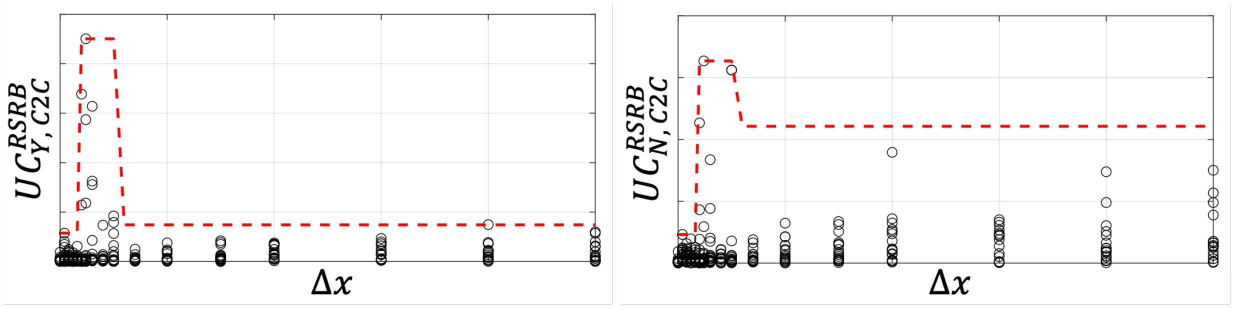


Fig. 9 Point-by-point comparisons between FUN3D and OVERFLOW simulations (in black dots) and a 99.7th percentile bound that defines the code-to-code uncertainty for the right booster side and normal force coefficients, both as a function of Δx.

Readers will note that a small number of points define a majority of the uncertainty space. These ”outliers” were points within the test set in which the plume chemistry was deemed more significant; as the code-to-code uncertainty primarily accounts for plume modeling differences in FUN3D relative to OVERFLOW, these outlying points were deemed critical to retain in the total uncertainty buildup.

B. Code-to-tunnel Uncertainty

The difference between simulated and experimentally observed booster separation physics is a critical component of the uncertainty model. The NASA Langley Unitary Plan Wind Tunnel (UPWT) conducted Test 1891 in 2014 on the Block 1 Crew vehicle [18]. The tunnel test provided uncertainty data for the Artemis I booster separation loads database as well, which supported the successful launch in Nov 2022; it is planned for use on the databases for the Artemis II and Artemis III launch vehicles. During this wind tunnel test, as depicted in Figure 10, the boosters and core stage were mounted on individual stings such that the three bodies could be moved independently of one another.

Air jets were also employed to approximate the plumes of the booster separation motors at several approximated thrust levels. However, due to the stings occupying the aft faces of the bodies, no engine plumes were established during the test. This is why the code-to-code uncertainty term focuses on this engine plume modeling fidelity.

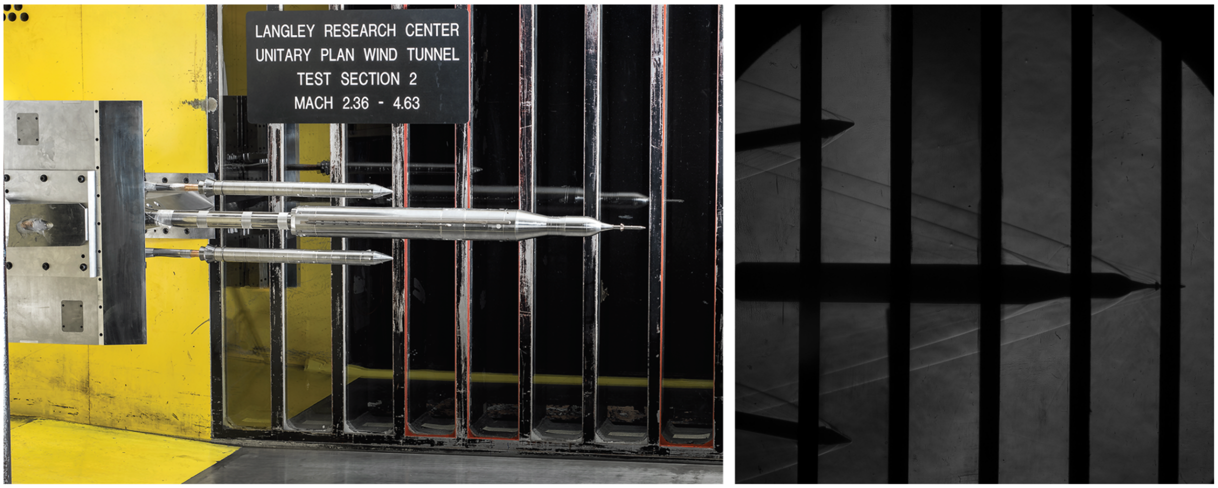


Fig. 10 Snapshot of the booster separation rig (left) and a Schlieren image (right) from the UPWT Test 1891.

Approximately 6,000 FUN3D simulations were conducted at several points throughout the wind tunnel test matrix. Booster separation motor stagnation pressures were matched to equate separation thrust coefficients. Statistics were taken separately for the separation motor-on and -off cases, with the former defining the uncertainty in the lower Δx range and the latter defining the uncertainty in the higher Δx range. Plots in Figure 11 detail the code-to-tunnel comparisons, as well as the 99.7th percentile bound, in both Δx ranges, for one of the 18 F&M coefficients.

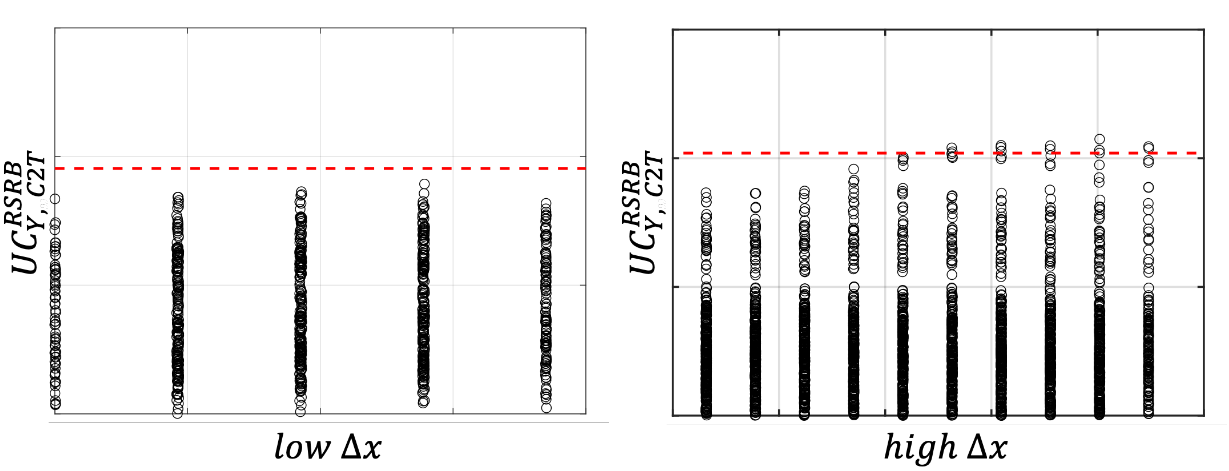


Fig. 11 Point-by-point comparisons between FUN3D and UPWT Test 1891 (in black dots) and a 99.7th percentile bound that defines the code-to-code uncertainty for the right booster side and normal force coefficients, both as a function of Δx, for both BSM-on (left) and BSM-off (right) cases.

C. Modeling Uncertainty

The modeling uncertainty quantifies the difference in represented physics by interpolating the CFD points rather than simulating additional CFD points. Given the sparsity of the CFD points, a fixed subset of interpolation test points is unlikely to be representative of the entire interpolated space. As such, the same cyclic-dropout procedure as used in the dimension flattening study, outlined in §II.C, was employed. A random 25% of the CFD data (evenly distributed in

Δx) was withheld for testing and the remaining 75% was treated as a mock database. This mock database was used to interpolate all 18 coefficients, against which values the withheld values were compared. The 99.7th percentiles of the interpolation errors, separately at each Δx, were recorded. This was done 50 separate times with different randomly withheld points such that all points were withheld roughly the same number of times.

Given the significant difference in point proximity between the two run matrices, interpolation errors were tracked separately for the two run matrices: the original points and the Sobol sequence points. When computing the interpolation errors for one data set, only that data set had points withheld for testing. This yielded two separate U_{mod} terms, which in turn yielded separate uncertainty models for the two regions of the data space. At any trajectory simulation point, the uncertainty model associated with the closest database point was applied.

Figure 12 presents the results of these interpolation tests, as a function of Δx, for the RSRB coefficients. Lines are drawn for the mean, maximum, and minimum interpolation errors out of the 50 dropout iterations. Note that all axes are presented on a logarithmic scale for better viewing. The notable difference in magnitude between these two interpolation error trends supports the decision to deliver separate uncertainty models. In both cases, U_{mod} was taken to be the mean (solid) trends in interpolation error.

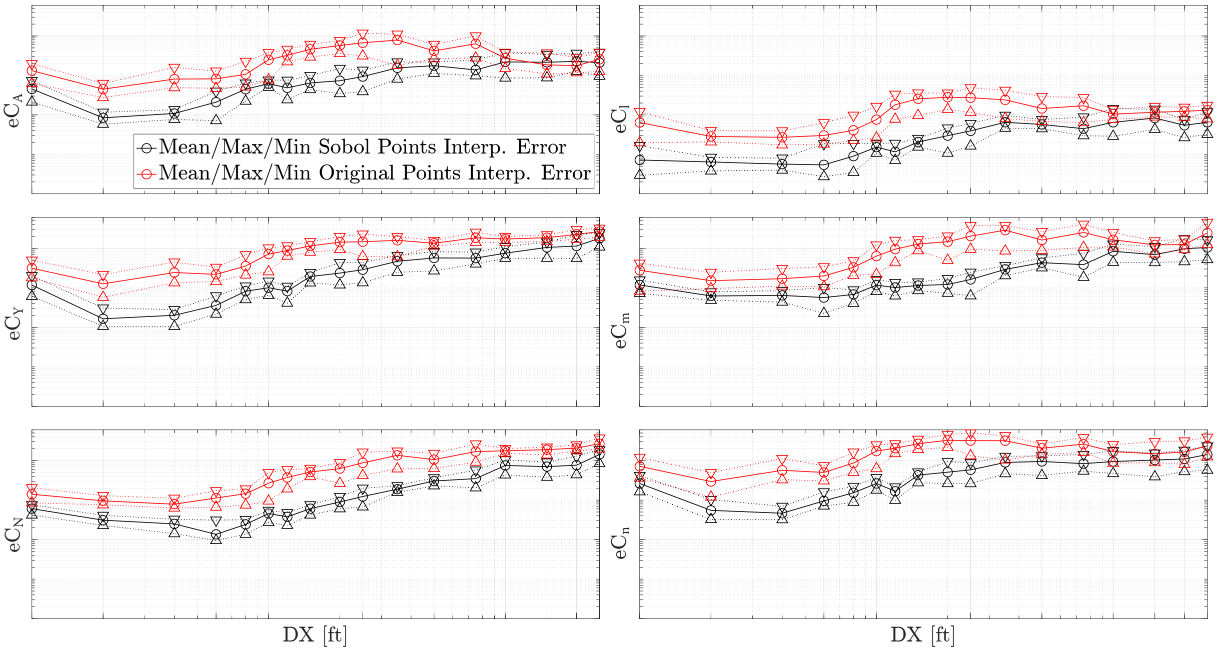


Fig. 12 Modeling uncertainties for the RSRB coefficients, computed separately for the two CFD run matrices. Mean trends are plotted as solid lines with circular.

IV. Booster Separation Trajectory Simulations

Once the booster separation F&M database is constructed, the impact of the aerodynamics on the booster separation event must be characterized. This verifies the aerodynamics model by integrating it into the larger SLS system. It also enables risk mitigation such that booster separation does not threaten the mission or its crew.

CLVTOPS is a multibody dynamics simulation tool that was developed expressly for the purpose of modeling liftoff and separation events. CLVTOPS is a complete toolchain for multibody dynamics simulations that spans all aspects of the data space, including organizing input data, executing the simulation, and post-processing and visualization of the results. A more complete description of the tool, along with some example data products, is provided in Ref. [19]. At the heart of CLVTOPS is the NASA-developed tool TREETOPS [20], which was released as part of the NASA Technology Transfer Program. TREETOPS handles the core of the dynamics simulation by formulating equations of motion for the SLS using Kane's method [21]. CLVTOPS is a multifunctional TREETOPS wrapper that enables, among other utilities, control of the forces, moments, and masses fed to TREETOPS and the collection of TREETOPS outputs such that trajectories can be iterated within a single execution. Output visualization is also a part of the CLVTOPS functionality. Additionally, CLVTOPS provides the capability to run Monte Carlo dispersions, which are used along with order statistics to provide confidence bounds on vehicle performance. For a detailed treatment of the statistical foundations of the CLVTOPS procedure for dispersed trajectories, see Ref. [22].

CLVTOPS enables an assessment of the SLS aerodynamic sensitivity by running simulations both with and without the booster separation aerodynamics enabled. Although CLVTOPS performs dynamic (viz. time-varying) trajectory simulations, the booster separation loads database – and indeed most aerodynamic databases for the SLS – is based on steady-state vehicle conditions. The validity of this assumption has been substantiated by several internal studies since the SLS program inception. Two sets of 2,000-run Monte Carlo simulations of the booster separation event were thus run, which yielded trajectories for the core stage and each SRB. Note that the aerodynamics model is only one of many models impacting the SLS during the booster separation event; CLVTOPS also includes core and SRB control activity, slosh, core and SRB engine performance, mass properties, sensor models, guidance, navigation, and other components that drive SLS behavior but are out of the scope of this paper.

Among many inputs, however, the aerodynamics model does have a substantial effect on the booster separation event and is often a driving factor in the separation analysis. As illustrated in Figure 13, the inclusion of the aerodynamic model greatly expands the separation drift envelope for the SRBs. When the aerodynamic F&M database is not applied, the SRB drift envelopes are relatively narrow, with no excursions far beyond the nominal case. Once the aerodynamics model is included, however, the drift becomes far less regular; this effect is particularly apparent when viewing the drift with respect to the y-z plane. As the SRB drift envelopes expand, SRB-to-SRB and SRB-to-core recontact becomes more likely, which endangers both mission and crew. Therefore, an accurate model of the aerodynamics is critical to ensure safe booster separation. In the plotted scenario, although they get closer to one another when aerodynamic effects are accounted for, no SRB-to-SRB contacts were observed to occur within the dispersion 3σ bounds.

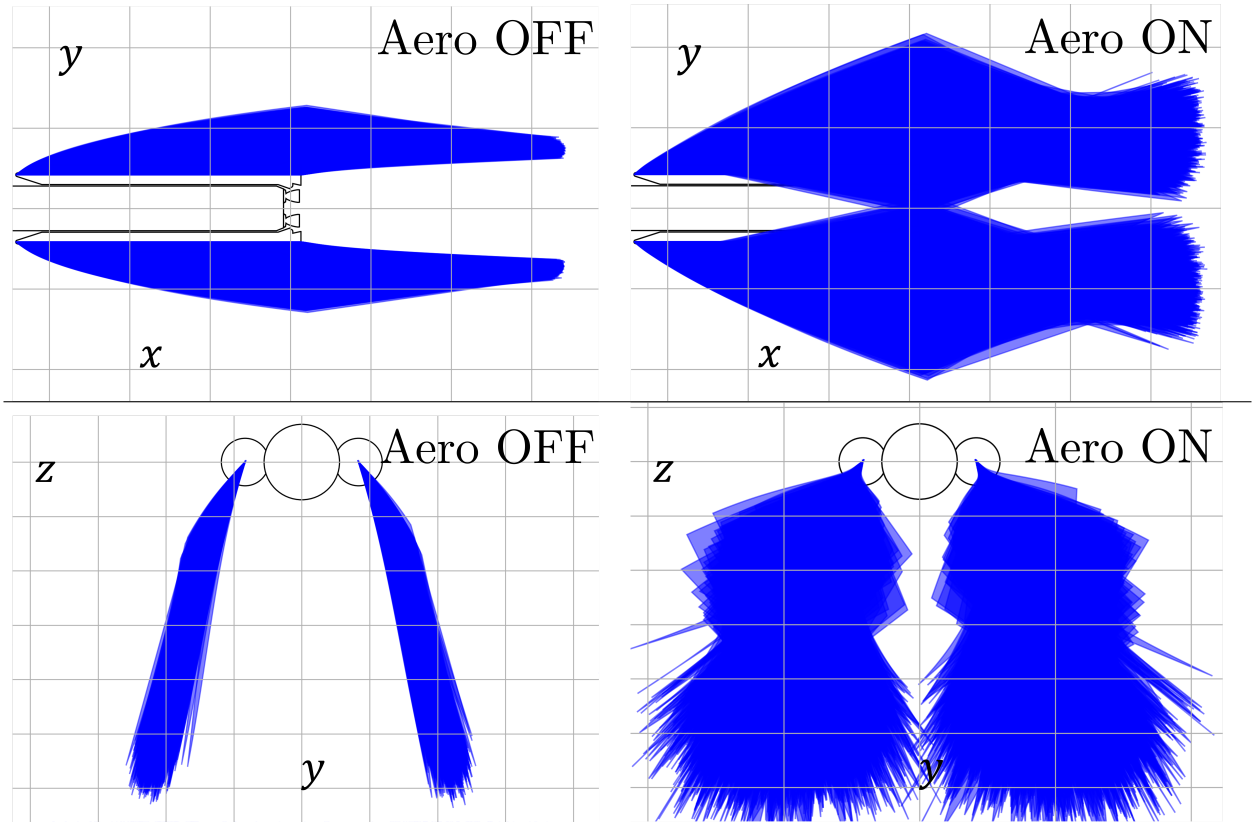


Fig. 13 Booster separation centerline drift envelopes, viewed in the x-y (top) and y-z (bottom) planes, with the aerodynamic F&M database disabled (left) and enabled (right).

V. Summary

The SLS booster separation event is a complicated and critical aspect of its launch missions. From an aerodynamic perspective, the core stage engine plumes, dwindling booster plumes, and separation motor plumes interact within supersonic freestream conditions in ways that are difficult to anticipate but necessary for risk mitigation. From a data science perspective, the problem exists within a highly irregular multi-input multi-output data space, only one part of which is the aerodynamic modeling effort. The process of accurately characterizing the aerodynamic behavior is challenging; this paper has detailed how that challenge has been met for the Artemis II launch vehicle.

The booster separation aerodynamic database development process has been detailed. An encompassing matrix of CFD simulations was executed, whose outputs were used to identify a more refined trajectory window in which additional data was necessary. After subsequent simulations were performed, a parameter sensitivity analysis was conducted to identify how the data space could be compressed without sacrificing fidelity. An uncertainty model was then developed, which was enabled by a wind tunnel test and additional computational efforts. The uncertainty model was designed to account for how the plume physics were modeled, how the simulations compared to physical observations, and how the database model itself represents the simulated fluid physics everywhere within the known data space. It was then shown how the resulting database informed booster separation trajectory dispersions, and how influential the aerodynamics in particular were to the separation event behavior.

Work is ongoing to develop similar booster separation aerodynamic databases for future SLS configurations. Additional parameter space sensitivity analyses are being conducted, novel database modeling strategies are being explored, and uncertainty audits are being planned to ensure that the SLS booster separation events remain a safe, reliable, and awe-inspiring aspect of future Artemis mission launches.

Acknowledgments

Appreciation is extended to H. Houlden (ViGYAN Inc.) and D. Chan (NASA LaRC) for their development of previous SLS booster separation databases and their guidance during the present development cycle. Acknowledgment is extended to P. McDonough for his instrumental role in orchestrating the booster separation simulation. Thanks are also given to S. Rogers, J. Meeroff, and A. Burkhead (NASA ARC) for their support in developing and maintaining the sizeable CFD simulation workflow.

References

[1] Rogers, S. E., Dalle, D. J., and Chan, W. M., "CFD Simulations of the Space Launch System Ascent Aerodynamics and Booster Separation," *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 0778.

[2] Dalle, D. J., Rogers, S. E., Lee, H. C., and Chan, W. M., "Inviscid and Viscous CFD Analysis of Booster Separation for the Space Launch System Vehicle," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0797.

[3] Biedron, R. T., Carlson, J., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Lee-Rausch, E. M., Nielson, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., "FUN3D Manual: 13.1," Tech. Rep. TM-2017-219580, NASA, 2017.

[4] Spalart, P. R., and Allmaras, S. R., "One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, No. 1, 1994, pp. 5–21.

[5] "ANSA Pre Processor," `https://www.beta-cae.com`, 2021. BETA CAE Systems, Version 20.1.1.

[6] Aftosmis, M., Berger, M., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," *38th Aerospace Sciences Meeting*, 2000. AIAA Paper 2000-0808.

[7] "AFLR3 Unstructured Grid Generator," `http://www.simcenter.msstate.edu/`, 2017. SimSys Software, Version 16.27.3.

[8] "Computational Aerosciences Productivity and Execution (CAPE) Toolbox," `https://github.com/nasa/cape`, 2022. Version 1.0.

[9] Renka, R. J., "Algorithm 661: QSHEP3D: Quadratic Shepard Method for Trivariate Interpolation of Scattered Data," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 14, No. 2, 1988, pp. 151–152.

[10] Nichols, R. H., Tramel, R. W., and Buning, P. G., "Solver and Turbulence Model Upgrades to OVERFLOW2 for Unsteady and High-Speed Applications," *36th AIAA Fluid Dynamics Conference*, 2006. AIAA Paper 2006-2824.

[11] Buning, P. G., and Pulliam, T. H., "Cartesian Off-Body Grid Adaption for Viscous Time-Accurate Flow Simulation," AIAA Paper 2011-3693, Jun. 2011.

[12] Murman, S., Chan, W. M., Aftosmis, M. J., and Meakin, R. L., "An Interface for Specifying Rigid-Body Motions for CFD Applications," AIAA Paper 2003-1237, Jan. 2003.

[13] Suhs, N., Rogers, S., and Dietz, W., "PEGASUS 5: An Automated Pre-Processor for Overset-Grid CFD," *AIAA Journal*, Vol. 41, No. 6, 2003, pp. 1037–1045.

[14] Meakin, R. L., "Automatic Off-Body Grid Generation for Domains of Arbitrary Size," AIAA Paper 2001-2536, Jun. 2001.

[15] Chan, W. M., Gomez, R. J., Rogers, S. E., and Buning, P. G., "Best Practices in Overset Grid Generation," AIAA Paper 2002-3191, Jun. 2002.

[16] Chan, W. M., "Developments in Strategies and Software Tools for Overset Structured Grid Generation and Connectivity," AIAA Paper 2011-3051, Jun. 2011.

[17] Meakin, R. L., "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," AIAA Paper 2001-2537, Jun. 2001.

[18] Wilcox, F., Pinier, J. T., Chan, D. T., and Crosby, W. A., "Space Launch System Booster Separation Aerodynamic Testing at the NASA Langley Unitary Plan Wind Tunnel," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 0796.

[19] Burger, B. S., Addona, C., Diedrich, B., Harlin, W., McDonough, P., Muscha, Z., Sells, H., and Tyler, D., "Space Launch System Liftoff and Separation Dynamics Analysis Tool Chain," *AIAA Scitech 2021 Forum*, 2021, p. 0822.

[20] "TREETOPS," , MFS-33566-1. URL `https://software.nasa.gov/software/MFS-33566-1`.

[21] Kane, T. R., and Wang, C., "On the Derivation of Equations of Motion," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 13, No. 2, 1965, pp. 487–492.

[22] Hanson, J. M., and Beard, B. B., "Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Verification," *Journal of Spacecraft and Rockets*, Vol. 49, No. 1, 2012, pp. 136–144.