

# Integration of Engine, Plume, and CFD Analyses in Conceptual Design of Low-Boom Supersonic Aircraft

Wu Li<sup>1</sup>, Richard Campbell<sup>2</sup>, and Karl Geiselhart<sup>3</sup>

*NASA Langley Research Center, Hampton, Virginia 23681, USA*

Elwood Shields<sup>4</sup>

*Alliant Techsystems, Inc., Hampton, Virginia 23681, USA*

Sudheer Nayani<sup>5</sup> and Rajiv Shenoy<sup>6</sup>

*Analytical Services & Materials, Inc., Hampton, Virginia 23666, USA*

**This paper documents an integration of engine, plume, and computational fluid dynamics (CFD) analyses in the conceptual design of low-boom supersonic aircraft, using a variable fidelity approach. In particular, the Numerical Propulsion Simulation System (NPSS) is used for propulsion system cycle analysis and nacelle outer mold line definition, and a low-fidelity plume model is developed for plume shape prediction based on NPSS engine data and nacelle geometry. This model provides a capability for the conceptual design of low-boom supersonic aircraft that accounts for plume effects. Then a newly developed process for automated CFD analysis is presented for CFD-based plume and boom analyses of the conceptual geometry. Five test cases are used to demonstrate the integrated engine, plume, and CFD analysis process based on a variable fidelity approach, as well as the feasibility of the automated CFD plume and boom analysis capability.**

---

\* AIAA Paper 2009-1171, 47th AIAA Aerospace Sciences Meeting, January 2009, Orlando, Florida.

<sup>1</sup> Senior Research Engineer, Aeronautics Systems Analysis Branch, Mail Stop 442, AIAA Member

<sup>2</sup> Senior Research Engineer, Configurations Aerodynamics Branch, Mail Stop 442, AIAA Associate Fellow

<sup>3</sup> Aerospace Engineer, Aeronautics Systems Analysis Branch, Mail Stop 442

<sup>4</sup> Senior System Engineer, Aeronautics Systems Analysis Branch, Space Division

<sup>5</sup> Senior Scientist, Analytical Services & Materials, 107 Research Drive

<sup>6</sup> CFD Engineer, Analytical Services & Materials, 107 Research Drive

## Nomenclature

$a$	=	speed of sound
$A$	=	area
$C_p$	=	pressure coefficient
$dp/p$	=	overpressure or pressure change
$K_{noz}$	=	ratio of the ideal nozzle area ratio (exit area / throat area) to the actual nozzle area ratio
$M$	=	Mach number
$P$	=	pressure
$q$	=	dynamic pressure
$T$	=	temperature
$x,y,z$	=	coordinates of points in space
$\alpha$	=	plume boundary angle
$\beta$	=	nozzle boattail angle
$\delta_{init}$	=	initial plume turning angle
$\gamma$	=	ratio of specific heats
$\theta$	=	nozzle internal half angle
$\nu$	=	Prandtl-Meyer angle
$\rho$	=	density
$\infty$	=	ambient condition

## Abbreviations

ADP	Aerodynamic design point
CFD	Computational fluid dynamics
CFL	Courant Friedrich Levy
HPC	High-pressure compressor
HPT	High-pressure turbine
HSR	High Speed Research
LPT	Low-pressure turbine
MFTF	Mixed-flow turbofan
NPR	Nozzle pressure ratio ( $P_t/P_{\infty,local}$ )
NPSS	Numerical Propulsion System Simulation
NURBS	Non-uniform rational B-splines
OML	Outer mold line
SFC	Specific fuel consumption
SLS	Sea level static
TOC	Top of climb
WATE	Weight Analysis of Turbine Engines

## I. Introduction

THIS paper documents an integration of engine, plume, and computational fluid dynamics (CFD) analyses in the conceptual design of low-boom supersonic aircraft. A variable fidelity approach is used. In particular, Numerical Propulsion System Simulation (NPSS) [1] is used for propulsion system cycle analysis and nacelle outer mold line (OML) definition. Also, a low-fidelity plume model is developed for plume shape prediction based on the NPSS engine data and nacelle geometry. This model provides a capability for the conceptual design of low-boom supersonic aircraft that accounts for plume effects. Then a newly developed process for automated CFD analysis is

used for CFD plume and boom analyses of the conceptual geometry. The automated CFD process consists of the following steps: (i) start from a *conceptual* definition of a complete aircraft, (ii) convert the aircraft geometry to a watertight CFD geometry in VGRID [2][3] (an unstructured grid generation code) format, (iii) generate sources for grid generation based on a few intuitive control parameters, (iv) use VGRID for unstructured grid generation, (v) use SSGRID [4] to shear and stretch the volume grid for improved boom predictions, and (vi) run USM3D [5][6] (an unstructured flow solver) for CFD solutions, including engine simulation based on NPSS engine data. The CFD pressure distributions can be used by PCBOOM [7] for sonic boom predictions, and the CFD plume analysis results can be used to calibrate the low-fidelity code for plume shape prediction. The current function of CFD analysis is to provide a higher fidelity prediction of the low-boom characteristics of the conceptual design, but the ultimate goal is to integrate the CFD analysis results into the conceptual design of low-boom supersonic aircraft.

There are three main objectives for this integrated engine, plume, and CFD analysis process: (i) demonstration of the feasibility of analysis automation from conceptual geometry to plume and boom predictions (for integration of high-fidelity analyses in the early stage of conceptual design), (ii) calibration of a low-fidelity plume model using CFD plume analysis results (for development of variable fidelity low-boom design methods), and (iii) understanding the significance of the plume shape on the ground signature (for possible mitigation of sonic boom using plume shape).

The primary motivation for this research effort is the need to use high-fidelity CFD analysis early in the conceptual design of low-boom supersonic configurations. Conceptual aircraft design usually utilizes linearized-theory aerodynamic codes to provide a relatively quick overall system performance evaluation in the early stages of the development of the concept. However, the low-boom characteristics of supersonic configurations are highly sensitive to aircraft component volume and lift distribution. Unlike a design for subsonic cruise efficiency, where the design of an isolated wing is a viable approach, a realistic low-boom supersonic concept must consist of all aircraft components including not only the wing but also the fuselage, nacelles, pylons, tails, and canards. Moreover, as a result of the level of geometric detail that is required in the design of a low-boom supersonic configuration, the exploration of a variety of drastically different configurations while using CFD analysis is necessary to check whether a given configuration has a low-boom ground signature. These requirements beg for automated CFD analyses of supersonic concepts with nacelles and tails that can provide accurate prediction of pressure distributions in the mid-field (3 to 10 body lengths below the aircraft). In addition, the engine's plume could have a significant

effect on ground signature, so understanding whether plume shaping is a viable approach for boom mitigation is important. As a result, an engine simulation capability becomes necessary to understand the trade-offs between engine performance and plume shaping for boom mitigation, which motivates the inclusion of plume simulation in the automated CFD analysis process.

Developing automated CFD analysis capability alone (without plume and boom predictions) is a daunting task. In the article “Translation Time” (*Mechanical Engineering*, October 2006), Jean Thilmany wrote that “... *analysts at the [Sandia] lab spend around 75 percent of their time cleaning up geometry after it's been imported from computer-aided design to analysis software.*” The article provides a telling story about one of the major bottlenecks in simulation-based design processes. In general, conceptual aircraft geometry is not as well-defined as CAD geometry. In particular, a conceptual aircraft geometry model usually does not define the intersection curves for the connecting surfaces.

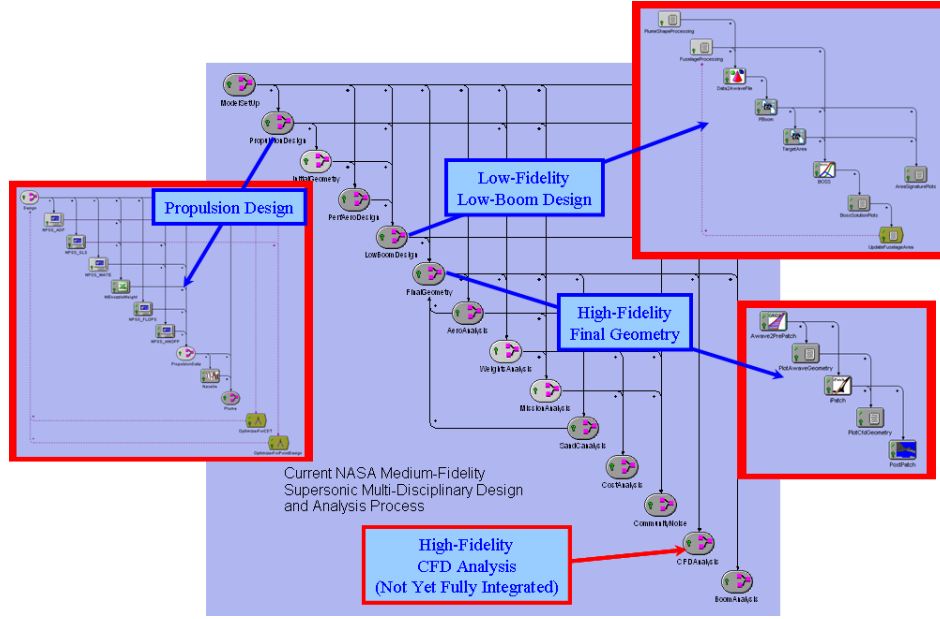
To overcome one of the barriers to the automation of CFD analysis, two computer codes were developed: iPatch (for converting conceptual geometry to CFD geometry) and AutoSrc (for automated line and volume source placements). These two codes close the gap between conceptual geometry and CFD grid generation. The process has been successfully tested on several supersonic concepts. The time required to set up from a supersonic concept to the input files for Euler CFD analysis is less than 20 minutes (e.g., setting up a few control parameters, clicking a few buttons in the analysis framework (Figure 1), and transferring files between desktop computers and clusters), with a turnaround time of a few hours, depending on the availability of computing hardware and grid resolution requirements.

Being able to generate a CFD surface solution is not enough for boom prediction, which requires accurate  $dp/p$  distributions at off-body locations. Historically, obtaining accurate  $dp/p$  distributions at off-body locations is a challenging CFD analysis task that usually requires special expertise and dedicated team efforts. During the Shaped Sonic Boom Demonstrator (SSBD) project, Lockheed Martin's Euler prediction of the overpressure at around 100 ft below the aircraft shows a good agreement with flight measurement data. The resulting ground signature prediction by PCBOOM matches the measurement fairly well. The predictions of the initial overpressure and signature duration are quite accurate, with about a 20 percent overestimate of the maximum and minimum overpressures in the ground signature (see Fig. 5b in [8]). More comparison results of CFD predictions and flight data at approximately 100 ft below the aircraft for the F-5E can be found in [9]. A detailed CFD analysis of the F-5E using

FUN3D with grid adaptation is documented in [10], which includes a comparison of a FUN3D solution to the Northrop-Grumman CFD prediction of  $dp/p$  at 1.5 body lengths (or 75 ft) below the aircraft, as well as a comparison of a FUN3D solution to the flight data at 1.82 body lengths (or 91 ft) below the aircraft. Comparison of CFD predictions and flight data for the NASA F-15B with the Gulfstream's quiet spike is documented in [11]. In particular, Howe, Waithe, and Haering [11] show that a hybrid CFD method generates  $dp/p$  distributions that are in very good agreement with the flight data at 90 ft below the aircraft and at 220 ft away from the aircraft at an off-track angle of  $50^\circ$ . However, the hybrid CFD method is *more expensive* both in labor hours and computational resources [11]. Recently, an a priori grid adaptation code (SSGRID) developed by Campbell [4] allows accurate predictions of mid-field (3 to 10 body lengths) pressure distributions with a reasonable grid size. Comparisons of USM3D predictions and the flight data show that USM3D predictions of  $dp/p$  using SSGRID are in very good agreement with flight data at 1.2, 2.6, 4.8, and 7.5 body lengths below the F-15QS. The main advantage of the SSGRID approach is that it is easy to use and can be automated in CFD analysis for boom prediction. SSGRID enables automated CFD-based boom analysis in the integration process.

Most plume studies are done with isolated jets. The main emphasis of these studies is to simulate the flow physics correctly by using turbulence models. Only two studies [12][13] address plume analysis of a complete aircraft configuration (F-16XL) at subsonic speeds ( $M = 0.242$  to  $0.97$ ). These two studies compare CFD surface vortex, pressure, and  $C_p$  distributions with measured data. Associated engine parameters for the CFD analyses are free-stream altitude (ft), free-stream Mach number, inlet duct exit static temperature ( $^\circ\text{R}$ ), inlet duct exit static pressure (psia), inlet duct exit velocity (ft/sec), inlet duct exit Mach number, mixing plane total temperature ( $^\circ\text{R}$ ), and mixing plane total pressure (psia). However, neither of these two papers has any discussion on engine simulation results. The current engine simulation capability of USM3D, along with automated plume volume source placement by AutoSrc, enables automated plume analysis in the integration process.

Our present goal is to establish an automated CFD analysis process that is robust (i.e., always generating converged solutions), works for most configurations of the interest (with good-quality grids regardless of the complexity of the configuration), is useful to the design of low-boom supersonic concepts, and is suitable for novice users without a CFD background. If necessary, CFD experts can be consulted to check results and to modify the grids and control parameters for more “accurate” CFD solutions.



**Figure 1. ModelCenter process for multi-fidelity multidiscipline analysis and design of supersonic concepts.**

The ultimate goal is to have a button-clicking process in ModelCenter™ (see Figure 1 for a snapshot of the overall process) for multi-fidelity, multidiscipline design and analysis of supersonic concepts. All of these codes have been integrated in ModelCenter. However, the ModelCenter component for SSGRID is still in debugging mode. In this study, the execution of VGRID, SSGRID, and USM3D was done manually by typing scripted commands in desktop computers and NASA supercomputers, respectively.

The paper is organized as follows. Section II discusses NPSS integration, including the nacelle geometry definition based on the engine data. Section III describes the low-fidelity plume model. In section IV, the details regarding the automated CFD analysis process for a conceptual geometry are provided. Section V begins with a description of the baseline supersonic business jet concept and an overview of the low-fidelity analysis process. It includes the results of both the low-fidelity and the CFD-based boom analyses of the baseline concept with five different nacelle shapes and compares the low-fidelity and CFD ground signatures and plume shapes. Section VI contains the conclusions.

## II. Integration of Propulsion System Modeling

The Numerical Propulsion System Simulation (NPSS) [1] can realistically model the physical interactions that take place throughout an engine, thereby accelerating the concept-to-production development time and reducing the need for expensive full-scale tests and experiments. Like traditional codes, an NPSS engine model is assembled

from a collection of interconnected components and controlled through the implementation of an appropriate solution algorithm [1]. For the purposes of the current study, a one-dimensional model of the engine flow and structure is created. The degree to which the nozzle is over or underexpanded is controlled with the parameter  $K_{noz}$ , which is defined in the NPSS model to equal the ratio of the ideal nozzle area ratio (exit area/throat area) to the actual nozzle area ratio. Varying  $K_{noz}$  effectively changes the nozzle exit area and affects thrust. Both the low and high-fidelity plume models use the same exit and throat areas and flow properties at the nozzle throat to generate the plume shape. Note that the flow properties at the nozzle throat are constant over the entire throat area and for each of the plume models.

The following subsections provide more detail on how to generate NPSS engine data and nacelle geometry in ModelCenter by using the one-dimensional engine model. Note that the purpose of this study is not to conduct a rigorous propulsion system design for a supersonic business jet but rather to provide input for high- and low-fidelity analyses of an aerodynamically efficient and low-boom supersonic business jet. As such, community noise and cost considerations are not treated rigorously, and some liberties have been taken with regard to the inlet and nozzle design details.

## **A. Model Integration**

The NPSS analysis code is implemented in ModelCenter as a component plug-in, which consists of Java code that runs within the ModelCenter process, an NPSS factory that creates NPSS object servers, and one server for each instance of NPSS in the user's ModelCenter model. Once the NPSS model file is set up for ModelCenter, the plug-in is easy and intuitive to setup for a run and does not require any editing of the wrapper code to incorporate changes in the model. As a component plug-in, NPSS looks like any other component in ModelCenter. Multiple instances of NPSS can be present in a single ModelCenter analysis, and if needed, multiple ModelCenter sessions can be running at the same time. The GUI for the plug-in can be accessed by double-clicking on the component's icon or right-clicking on the icon and selecting from the menu.

## **B. Turbo-Machinery Modeling**

Turbo-machinery maps for the low- and high-pressure compressors are used in the engine performance modeling. These maps are based on the fan and high-pressure compressor (HPC) maps used during the High Speed

Research (HSR) Program to model the performance of the reference mixed-flow turbofan engine. Similarly, maps from the HSR Program reference engine are also used for the high- and low-pressure turbines. Turbo-machinery adiabatic efficiencies are chosen to be 86 percent for the fan, 85.65 percent for the HPC, 90 percent for the high-pressure turbine (HPT), and 94 percent for the low-pressure turbine (LPT), which are representative of HSR technology levels. The turbine cooling levels are estimated to be 12.3 percent of the HPC flow based on the turbine rotor and stator material temperature limits.

### **C. Installation Effects**

The inlet maps or overall performance characteristics are based on the results of an engineering analysis of an external compression half-round inlet described in [14]. The cruise inlet pressure recovery is 0.9448. Based on past experience, the lip angle of this inlet ( $\sim 13^\circ$ ) was reduced to  $6^\circ$  in order to improve the chances of achieving an acceptable sonic boom ground signature and reduce cruise drag. This reduction might have required a redesigned, possibly more complex mixed compression or less efficient inlet; however, subsequent low-fidelity analyses at higher lip angles have shown that the penalties are not as great as previously thought. The nozzle aft-body drag effects are not considered in this study. At the aerodynamic design point (ADP), the convergent-divergent nozzle is assumed to be fully expanded at all conditions with a velocity coefficient of 0.985 and is consistent with the mixer-ejector nozzle technology from previous HSR studies. While the nozzle boattail angle can have a significant impact on sonic boom and off-design performance (see subsection II.A), the lack of nozzle aft-body drag effects has little impact on the results of this study because the nacelle geometry is fixed at the ADP, and the effects of the boattail are included in the low- and high-fidelity analyses.

### **D. Weight and Flowpath Analysis**

Weight Analysis of Turbine Engines (WATE) is a computer code used to estimate the weight and dimensions of a gas turbine engine that was originally developed by the Boeing Military Aircraft Company in 1979 and is described in [15]. The computer code calculates the weight and dimensions of engine components primarily using a semi-empirical method augmented with analytical calculations for specific component elements. The mixer ejector nozzle weight is scaled from analysis results that were obtained during the HSR Program. The geometric data from the code is used to generate the low-fidelity nacelle geometry.



## E. Cycle Selection and Nacelle Geometry Definition

The cycle used for this study is a mixed-flow turbofan (MFTF). The maximum combustor exit temperature is 3,500 °R, and the compressor discharge is limited to 1,700 °R, which are representative of HSR technology levels. Because the focus of this study is on low-boom concepts, the MFTF is optimized to create an aerodynamically efficient nacelle and matches the thrust lapse [top of climb (TOC) thrust/sea level static (SLS) thrust] requirement of the aircraft, while minimizing the cruise specific fuel consumption (SFC). The cruise thrust and thrust lapse requirements are determined from preliminary analyses to be 7,000 lb and 0.3188, respectively, at the ADP of Mach 1.8 and an altitude of 52,000 ft. Given these constraints, fan and HPC pressure ratio and throttle ratio are optimized to minimize the cruise SFC. The results are shown in Table 1. The flowpath and weight results from WATE++ are shown in Figure 2 and Table 2.

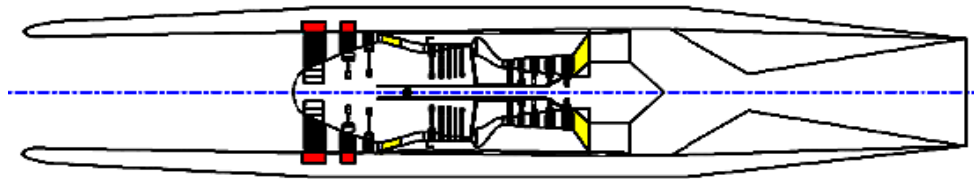


Figure 2. WATE++ flowpath.

Table 1. Cycle Parameters and Constraints

Fan pressure ratio	3.945
HPC pressure ratio	5.979
Throttle ratio	1.050
$T_3$ , °R	1,700
Lapse	0.3189
$T_4$ , °R	3,500
SFC, lb/hr/lb	1.091

ADP at Mach 1.8 and altitude of 52,000 ft

Table 2. WATE++ Analysis Results

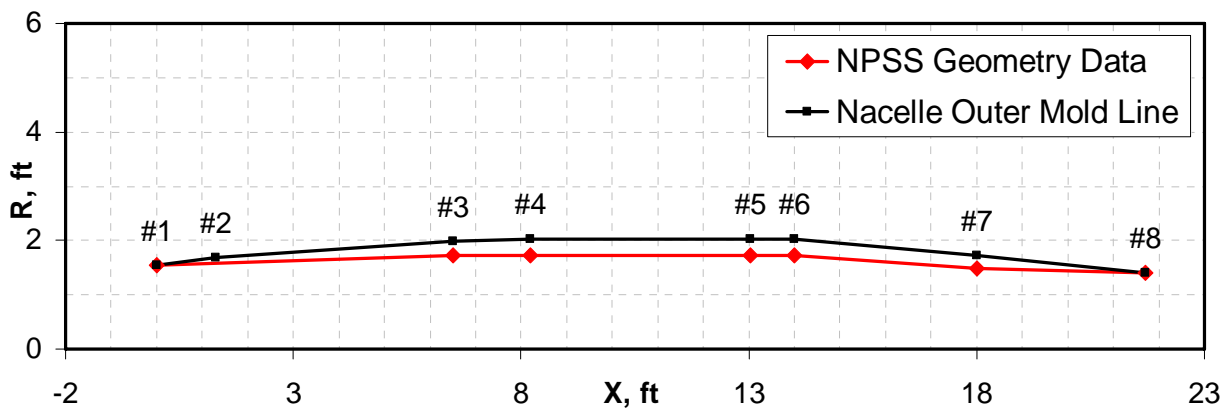
Bare engine weight, lb	3,467*
Inlet, lb	1,859
Mixer ejector nozzle, lb	2,871
Total installed weight, lb	8,197

\* Bare engine weight is WATE++ “bare engine weight” minus “engine mount weight” minus “fan exhaust cowl weight” (tailpipe) weight minus nozzle weight.

Note that the MFTF optimization is decoupled from the mission, noise, and takeoff/landing analyses because of the primary focus on low-boom and low-drag properties of the concept in this study. Therefore, the engine generated by the current optimization process is not necessarily the most desirable engine at the system level. For example, airport noise of the current engine might exceed the regulatory limit.

Because NPSS/WATE++ had no supersonic inlet design or mixer ejector nozzle design capabilities at the time of this study, the actual nacelle OML is based on turbo-machinery geometry and inlet and nozzle length to diameter ratios (L/D). The inlet L/D is 2.1 and the mixer ejector nozzle L/D is 2.5. The inlet capture area and nozzle exit and throat areas are known from cycle analysis. The propulsion data assembly in ModelCenter builds the nacelle geometry (see Figure 3) based on the following assumptions:

- #1 in Figure 3. At the inlet entrance, the radius is based on the capture area.
- #2 in Figure 3. At 20 percent of the inlet, the radius is based on an initial lip angle of six degrees.
- #3 in Figure 3. At the fan face, the radius is based on the fan diameter + containment + 15 percent.
- #4–6 in Figure 3. From the fan exit to the nozzle entrance, the radius is based on the fan diameter + containment + 17 percent.
- #7 in Figure 3. At the nozzle mid-point, the radius is the largest of the bypass duct tip radius + 17 percent and the average of the maximum and exit radii.
- #8 in Figure 3. At the nozzle exit, the radius is based on the nozzle exit area.



**Figure 3. Low-fidelity nacelle OML.**

The points #3 to 7 in Figure 3 are defined based on a 15 to 17 percent margin requirement for structural layout and aerodynamic efficiency of the nacelle shape, while the choice of lip angle is based on aerodynamic efficiency and sonic boom signature alone. This heuristic process to define the nacelle shape works well for the MFTF with

relatively low bypass ratios, but higher bypass ratios or other cycles may require additional design capability to generate a credible and aerodynamically efficient OML of the nacelle.

The nacelle OML and the throat area provide the complete nacelle geometry (shown in Figure 4) for USM3D engine simulation without inlet flow simulation.

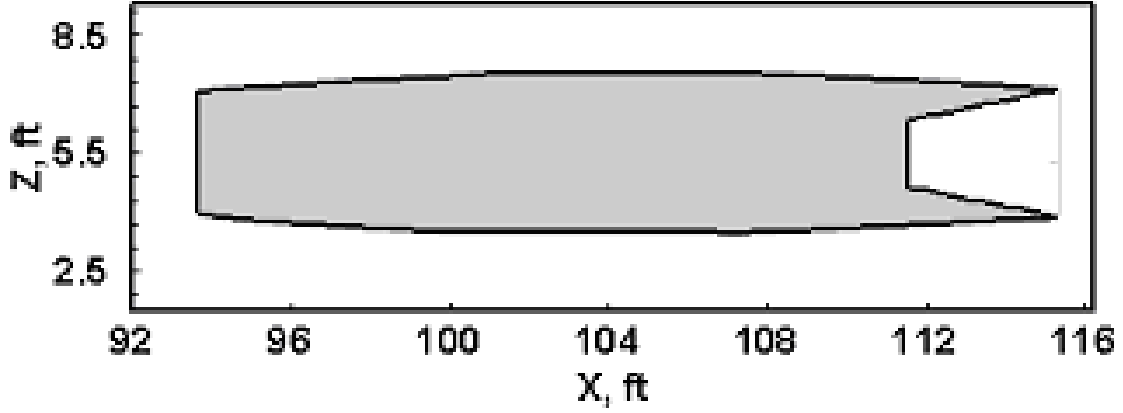


Figure 4. Nacelle geometry based on NPSS engine model.

### III. Low-Fidelity Prediction of Plume Shapes

The low-fidelity nozzle plume profile or boundary can be computed assuming one-dimensional ideal gas flow [16][17]. The implemented computer model uses a variable specific heat ratio ( $\gamma$ ) based on temperature and accounts for the effect of the nozzle boattail on the local static pressure by assuming Prandtl-Myer expansion. It includes factors to adjust how the initial “average plume slope” is determined, the degree to which the nozzle divergence half-angle affects the plume shape, and the degree to which the nozzle boattail angle affects the plume shape. The computer model produces underexpanded plume shapes that agree well with those presented in [17]. The inputs to the model are the exit and throat areas and the flow properties at the nozzle throat from NPSS and the nozzle internal and external flap lengths. The model begins by making the determination of whether the nozzle is over or underexpanded. First the ideal or fully expanded Mach number is determined from the stream-tube-area relation for isentropic perfect gas flow:

$$\frac{A}{A^*} = \frac{1}{M} \left( \frac{2 + (\gamma - 1)M^2}{\gamma + 1} \right)^{\frac{\gamma + 1}{2(\gamma - 1)}}$$

where  $A/A^*$  is the ratio of the nozzle exit area to the throat area from the NPSS simulation. The ideal pressure is then computed from the isentropic, perfect gas flow equation for static to total pressure ratio ( $P/P_t$ ) and the ideal Mach number. If the resulting pressure ( $P_e$ ) is greater than the local free-stream pressure  $P_\infty$ , which can be affected by the nozzle boattail (see Figure 6), then the nozzle is underexpanded. If  $P_e < P_\infty$ , it is overexpanded. For underexpanded nozzles, an oblique shock is created by the plume itself that affects the local free-stream pressure, creating the need for an iterative process. The following two subsections provide more details on the implemented computer model for prediction of underexpanded and overexpanded plume shapes.

### A. Underexpanded Nozzle Flow

A modified version of the method proposed by Nash, Whitaker, and Freeman [17] is used to predict the plume shape for underexpanded nozzles. The basic properties of a supersonic jet exhausting into a supersonic free-stream are illustrated in Figure 5. The basic approach is described in detail in [17], so it is only summarized here with emphasis placed on the differences. The initial plume boundary angle ( $\alpha_1$ ) is determined through iterations for solving the following equation:

$$A_1 = v_1 - v_{\text{noz}} + \theta$$

where  $v_{\text{noz}}$  is the Prandtl-Meyer angle that corresponds to the jet Mach number, and  $v_1$  is the Prandtl-Meyer angle that corresponds to the nozzle pressure ratio (NPR) ( $P_t/P_\infty$ ). The expansion of the nozzle flow creates an oblique shock that causes an increase in the free-stream pressure as seen by the jet ( $P_2$ ). The initial plume slope ( $\alpha_1$ ) is varied until the pressure at  $P_2$  reaches equilibrium. The method in [17] has been modified to account for a nozzle boattail angle ( $\beta$ ) and the ratio of the specific heats ( $\gamma$ ). The local pressure in the vicinity of the nozzle exit is reduced by the Prandtl-Meyer expansion of the flow around the nozzle boattail [18]. The ratio of specific heats is a function of the jet temperature and fuel-to-air ratio. The boattail angle can have a significant impact on the plume shape (see Figure 6); however, because the interaction of the free-stream flow-field in the presence of the nozzle boattail and the plume boundary is very complex and the actual effect of the boattail on the local pressure is unknown, the current method applies a correction to the free-stream pressure depending on the relative magnitude of  $\beta$  and  $\alpha_1$  and only when  $\beta > \alpha_1$ . As noted above, this is an iterative process that continues until the initial plume angle remains constant. Experimental data or validated CFD results are needed to calibrate the plume shape model.

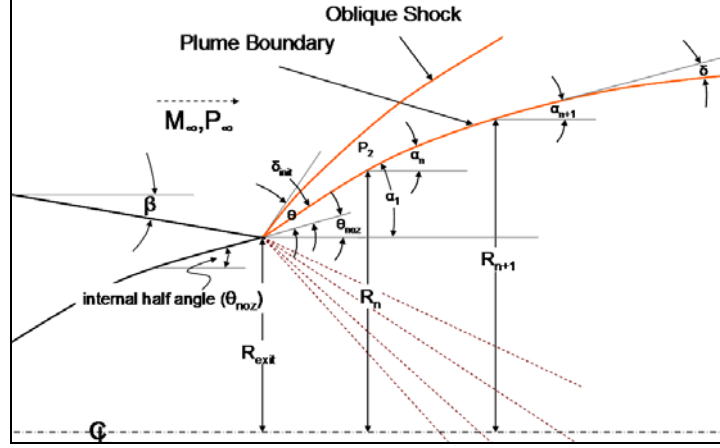


Figure 5. Underexpanded plume.

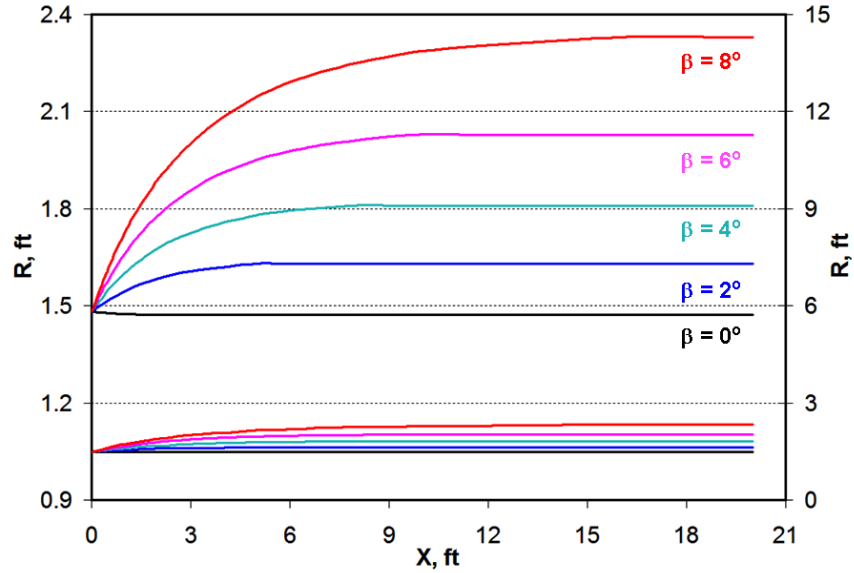


Figure 6. Effect of boattail angle (the lower curves are scaled to  $R: X = 1$ , shown on the secondary axis).

## B. Overexpanded Nozzle Flow

For overexpanded nozzles (Figure 7), an initial average pressure ( $P_{avg}$ ) is determined based on the free-stream pressure and the degree to which the nozzle is overexpanded. The initial free-stream turning angle  $\Delta v_1$  is computed from the Prandtl-Meyer expansion of the flow from  $P_\infty$  to  $P_{avg}$ . The nozzle flow is turned by  $\delta$  degrees through an oblique shock whose strength is determined from the pressure ratio ( $P_{ideal}/P_{avg}$ ). The axial component of the nozzle flow is then used to compute a static temperature and density from the isentropic, adiabatic, perfect gas flow equations. The nozzle exit velocity and Mach number are then calculated based on a conservation of mass. A new  $P_{avg}$  is then computed from the following equation:

$$P_{avg} \cong P_{avg_{initial}} \frac{P_2}{P_1}$$

where the static pressure ratio ( $P_2/P_1$ ) is computed from the oblique shock relation using the axial component of the Mach number and  $\theta$ , and a new nozzle exit static temperature is computed from the adiabatic, perfect gas flow equation. The plume shape is then determined by tracing the points on the plume boundary by incrementing through a fixed  $\Delta v$  until the Prandtl-Myer angle is zero. The initial jet Mach number is the axial component and is updated at each step based on a conservation of mass using the plume cross-sectional area. The rest of the jet flow properties are updated using the isentropic, adiabatic, perfect gas flow equations. Experimental data or validated CFD results are needed to calibrate the model.

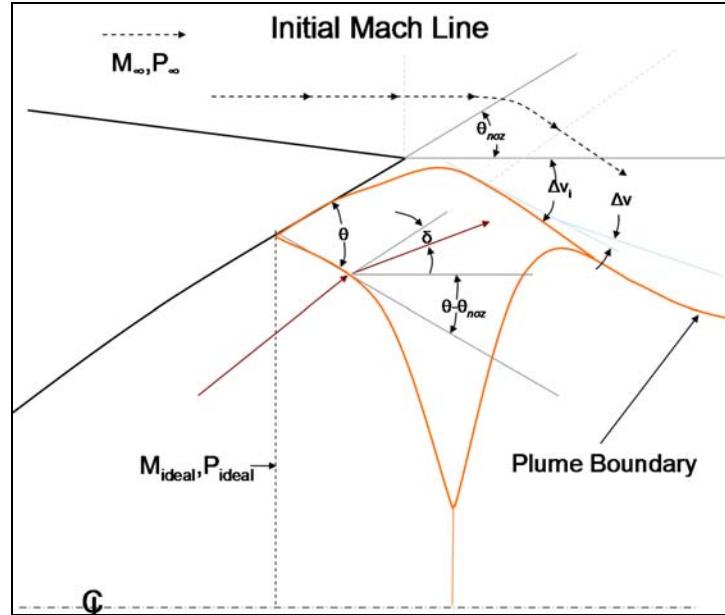


Figure 7. Overexpanded plume.

#### IV. Integration of CFD Analysis in Conceptual Design

The integrated CFD analysis process is completely automated and controlled by a few user input parameters. It starts from *conceptual* geometry of a complete supersonic aircraft, converts the aircraft geometry to a watertight CFD geometry in VGRID format, generates sources for grid generation based on a few intuitive control parameters, uses VGRID for unstructured grid generation, uses SSGRID to shear and stretch the volume grid for improved boom

prediction, and runs USM3D [6] to obtain a CFD solution. The following subsections provide more details on the geometry conversion, automated source placement, grid shearing and stretching, and USM3D engine simulation.

#### **A. iPatch: Conversion from Conceptual Geometry to CFD Geometry**

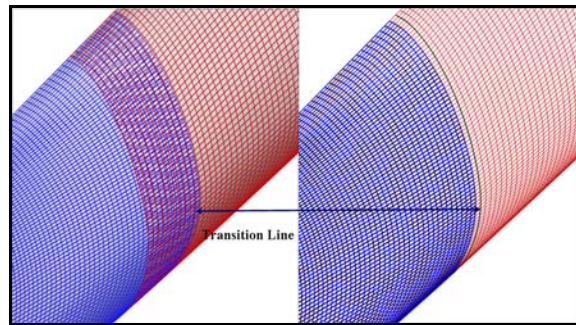
To convert the conceptual geometry to CFD geometry, a computer code called iPatch (for intelligently patching surface grids) was developed. The computer code iPatch does the following three tasks automatically: (i) use bicubic B-splines to extrapolate (if necessary) each surface in a conceptual geometry so that all of the independently defined geometry components (such as the wing and the fuselage) can be intersected to form a watertight CFD geometry, (ii) compute the intersection curves of surface patches at any resolution (up to  $10^{-4}$  accuracy) specified by the user, and (iii) write the B-spline surface patches and the corresponding boundary points for the watertight CFD geometry in the format that can be directly used by the grid generation tool VGRID.

The input conceptual geometry requirements for iPatch are the following:

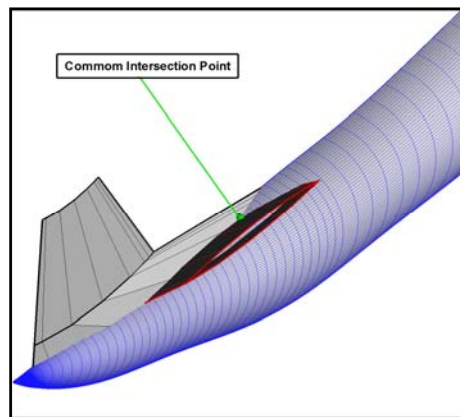
- (R1) Input geometry is in PLOT3D format [19]; each component surface is defined by a rectangular grid  $\{(x(i,j), y(i,j), z(i,j)): 1 \leq i \leq m, 1 \leq j \leq n\}$  that represents a smooth B-spline surface.
- (R2) All surfaces in the PLOT3D file conceptually represent a watertight geometry of components of an aircraft on the half-space  $y \geq 0$ .
- (R3) Overlapping surfaces are not allowed but could be fixed by a utility code “fixp3d.”
- (R4) Any two connecting surfaces shall have a “visually” common boundary curve or can be described by an intersection relationship that is defined in a geometry specification file. For example, if the first surface grid represents the fuselage and the second represents the inboard wing section, then users can specify the intersection matrix entry (2, 1) to be 1 for wing intersecting fuselage.
- (R5) The intersection of two surfaces can be at a “conceptual” level. That is, if some gap exists between the inboard wing section and the fuselage, iPatch uses extrapolation of the wing to close the gap and find the intersection curve of the extrapolated wing and fuselage. However, the intersection is directional (along either the  $i$  or  $j$  index direction), and each intersecting grid line (or its spine extrapolation) on the first surface should intersect the second surface.
- (R6) No two intersection relationships will result in a common intersection point of three surfaces.

The output files of iPatch are iges, d3m, and mapbc files that define the CFD geometry in VGRID format. The iges file gives the non-uniform rational B-splines (NURBS) definition of the OML of the geometry. The d3m file defines how the OML is broken into surface patches whose boundary curves are defined by points. The mapbc file specifies what the boundary condition is on each patch and the NURBS surface definition of each non-planar patch in the iges file.

The utility code fixp3d first finds the two grid lines on the two surface grids that are closest to each other in Hausdorff distance (a metric to measure the discrepancies of two sets); then fixp3d uses one of the grid lines as the transition line and extends grid lines on one grid to the other grid to form a merged grid. See Figure 8 for a merged grid from two overlapping grids.



**Figure 8. Fix overlapping grids by using a utility code fixp3d.**

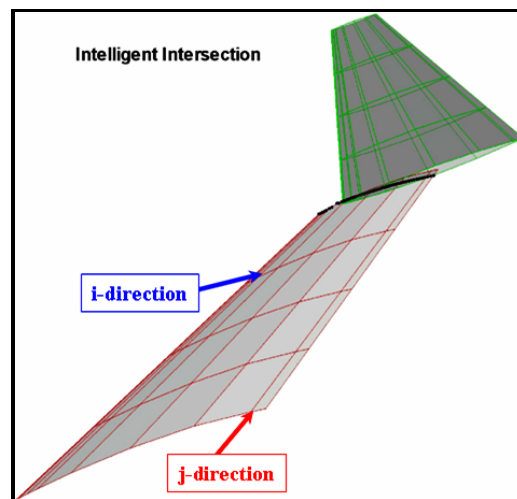


**Figure 9. Intersection of three surfaces at a common point.**

In theory, the only geometry limitation of the automated geometry conversion process is requirement (R6) listed above. Figure 9 shows an example of the fuselage, the lower part of vertical tail, and the pylon intersecting at a common point. All other requirements are for coding convenience. For example, in Figure 10, the vertical tail only partially intersects the horizontal tail, which violates (R5) because some of the grid lines, such as the leading edge of



the vertical tail, do not intersect the horizontal tail. The surface-to-surface intersection algorithm used by iPatch is efficient (i.e., it uses the Newton search direction and a local high-order search method based on the exact intersection of the extrapolated grid line and the tangent plane at the intersection point) and robust (i.e., it uses multiple search directions including the steepest descent direction and Gauss-Newton search direction to ensure the success of the line search). In this example, for each grid line in the vertical direction on the vertical tail, the intersection algorithm finds the point on the extrapolated curve of the grid line that is closest to the horizontal tail surface. For each extrapolated grid line that intersects the horizontal tail surface, two possible intersection points exist, one on the lower surface and one on the upper surface. The intersection algorithm is implemented to search for the intersection point that is closest to the mid-point of the original grid line. The strategy allows iPatch to find the correct intersection points for surface-to-surface intersection problems. In the case for which the extrapolated grid line does not intersect the horizontal tail, the intersection algorithm terminates with the highest point on the grid line that is closest to the horizontal tail.

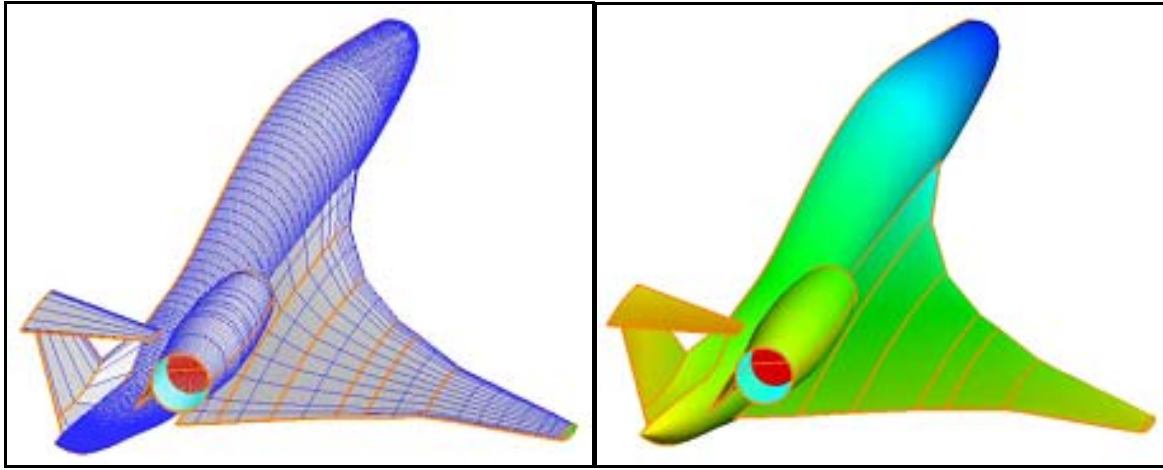


**Figure 10. Partial intersection of a vertical tail grid with a horizontal tail grid.**

The complication for the partial intersection case is the question of how to cover the open part of the vertical tail to form a watertight geometry. A planar patch can easily be used to cover the opening, but the analysis has to further break the patches for the vertical tail so that each boundary curve will be exactly shared by two patches (a requirement for VGRID inputs). Treating such a possibility makes the arrangement of all the patches and boundary curves in the proper order that is required by VGRID difficult, but it is doable.

The manual part of the conversion process involves the construction of a topology specification file that identifies the aircraft components defined by rectangular grids and the intersection relationships between grids. See

the sample topology specification file in the appendix for the conceptual configuration shown in Figure 11. The intersection operation tells iPatch which intersection curves needs to be computed. The intersection index tells iPatch how to calculate the intersection points: 1 means that each intersection point is generated by a grid line (with a constant  $j$ -direction parameter) of the first grid intersecting the second grid surface, while 2 means that each intersection point is generated by a grid line (with a constant  $i$ -direction parameter) of the first grid intersecting the second grid surface. In this case, four intersection problems exist: (i) wing and fuselage, (ii) pylon and fuselage, (iii) pylon and nacelle, and (iv) vertical and horizontal tails.



**Figure 11. Conceptual geometry (left) and the corresponding CFD geometry (right).**

In general, if the geometry file contains many grids, then constructing this topology specification file could be time consuming and prone to errors (e.g., a wrong intersection index). A function can be added in GridTool [20][21] to interactively define the grouping and intersection relationships. For the ModelCenter process, a simple pattern recognition algorithm was developed for automated generation of this topology specification file. In theory, given an aircraft geometry in PLOT3D format, aircraft components such as fuselage, wing, winglet, wingtip, pylon, nacelle, canard, vertical tail, and horizontal tail, as well as the intersection relationships, can be automatically identified with some minor restrictions on the input geometry grids. The converted CFD geometry is also shown in Figure 11. Note that the gap between wing and fuselage is filled by extrapolating the inboard wing section toward the fuselage, and all of the surfaces are trimmed by the intersection curves to form a watertight CFD geometry.

iPatch uses one input parameter to control the conversion process: the resolution of boundary curves in the d3d file. Because VGRID requires a discrete definition of boundary curves for the watertight CFD geometry, iPatch provides a global density parameter to control the number of points in the discrete definition of a curve. The global density parameter is defined in terms of the number of points per unit length. To prevent an unreasonable choice of

this density parameter, a minimum of 16 and a maximum of 500 points are used to define one discrete curve. iPatch uses an adaptive method for defining the points on a curve so that the turning angle between two consecutive line segments is no less than  $175^\circ$ , which ensures that any three consecutive points on a curve cover a relatively flat segment of the curve so that the approximation error between the discrete and continuous curves is very small.

Note that if one has parametric definitions of some or all aircraft components, then the surface grids (at any resolution) can be relatively easily generated in PLOT3D format, along with the corresponding topology specification file. This allows users to decouple the geometry parameterization from the geometry conversion process. Moreover, the current iPatch interface allows users to insert a new parametric geometry model for a particular component (such as wing) into the concept design process with minimum effort to integrate this new parametric component with other aircraft components for CFD analysis.

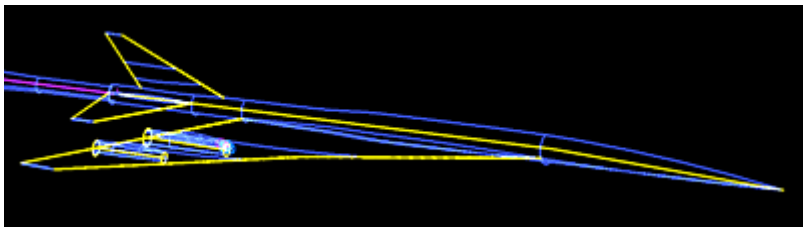
## **B. Automated Source Placement and Volume Grid Generation**

The iPatch code writes all of the model surface geometry information, as well as an outer flow-field boundary box definition, to a file in VGRID input format. At this point in the input preparation process, a user would then manually position and specify the size of sources along the geometry and in the field using the GridTool [20][21] graphical user interface. These sources are used by VGRID to determine grid spacing on the surface and throughout the field using initial spacings along with growth-rate equations [22]. This has typically been one of the more time-consuming parts of the grid-generation process, as a complex configuration could easily require hundreds of sources to obtain a good grid.

The AutoSrc code was developed to provide an automated, knowledge-based approach on how to place and size the sources for both novice and experienced users. The code uses specific component names to locate key geometry curves for the associated patches and then uses this information to determine source location and size. The list of components currently available in AutoSrc includes: fuselage, nose, wing, canard, horizontal tail, vertical tail, nacout1, pylon1, nacout2, and pylon2, where indices 1 and 2 allow surface patches for up to two engine nacelles. For lifting surface components such as the wing and tails, sources are automatically placed along the leading and trailing edges of the components, with a user option for additional line sources along span-wise lines at intermediate constant chord locations. In addition, anisotropic stretching values can be input to increase the grid cell size in the spanwise direction and reduce the number of cells required. The user also inputs values for the size of cells at the

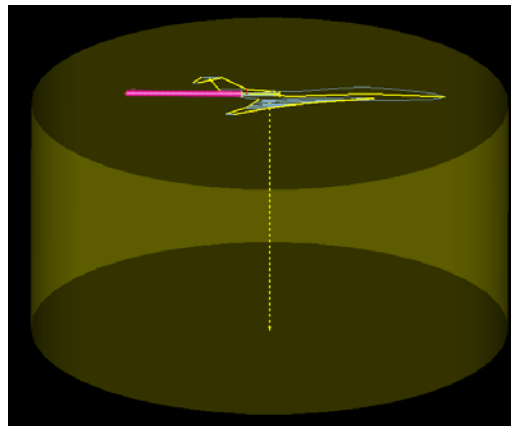
leading and trailing edges relative to cells in the mid-chord region of the lifting surfaces. Suggested values for these input variables, based on feedback from researchers with extensive experience with VGRID, are included for novice users.

For the fuselage and nose sources, the sizing is based on body length, and the location is determined by dividing the fuselage into four segments and placing the sources along the centerline of each segment. An optional sting source extending downstream from the end of the fuselage can be specified for cases that represent wind-tunnel models. The source layout for nacelles includes a ring of line sources around the inlet and exit, as well as a stream-wise source down the centerline of the nacelle.



**Figure 12. VGRID line sources automatically generated by AutoSrc.**

The line sources (see Figure 12) are used to determine the size of the surface and the field grid in the vicinity of each source. AutoSrc also includes an overall scaling option that multiplies each source by a given factor to allow the user to produce consistently coarser or finer grids for a given geometry. A pair of growth-rate factors used in the latest version of VGRID to determine field grid density is automatically set by AutoSrc.



**Figure 13. VRID volume sources for plume and boom analysis.**

In addition to the basic layout of line sources that are described herein, two options that use the new volume source capability in VGRID [22] have been included in AutoSrc. The first is used for computation of the sonic boom signature of an aircraft using the mid-field pressure distribution from CFD and the propagation code PCBOOM, where a cylindrical volume source is automatically sized and located under the configuration to provide a dense field

grid for resolving shocks and expansions (see the cylinder around the dashed center line in Figure 13). This grid is further modified by the SSGRID code, which is described in the next subsection, to help reduce dissipation in the flow solution. The second option places cylindrical sources streamwise behind the nacelle exits to create a sufficient grid to resolve the engine plume shape for powered simulations (see the pink cylinder in Figure 13). Currently, the length of the plume source is a user input, with a recommended value of three nacelle lengths based on limited experience.

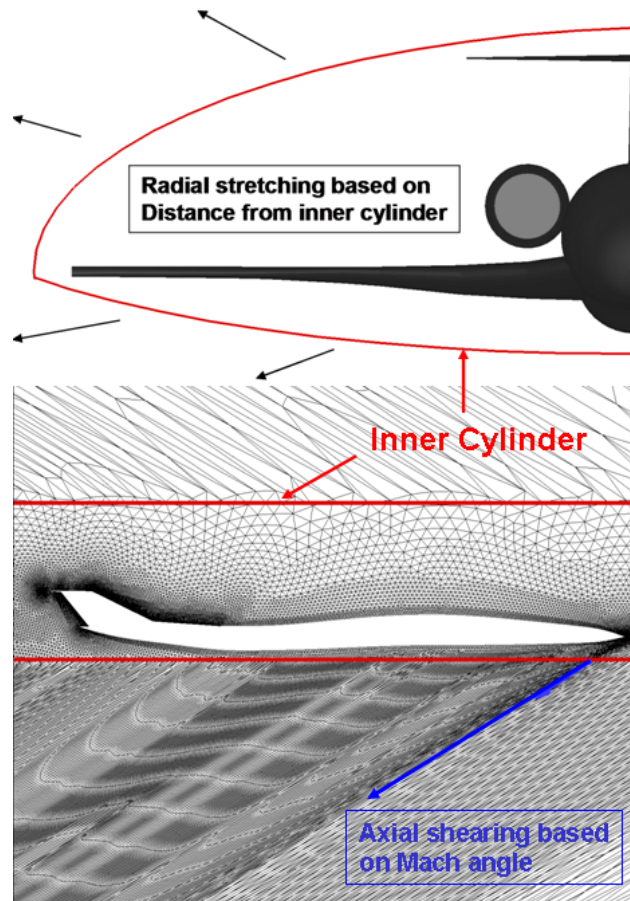
Although most of the grids that have been generated using AutoSrc coupled with iPatch have been for inviscid flow computations related to sonic boom, the code also will automatically determine the parameters that are needed by VGRID for full-viscous or wall-function grids based on Reynolds number and reference length input by the user, with other parameters set to default values based on recommendations from experienced users.

### **C. Grid Shearing and Stretching for Boom Prediction**

The SSGRID code uses an a priori grid adaptation approach, where the initial grid is clustered and oriented based on general flow characteristics that are known in advance of running the flow solver. Clustering surface grid points near the leading edge of a wing to resolve both the surface curvature as well as the flow acceleration is a common example of this knowledge-based approach used in both unstructured and structured grid generation. Control of the field grid characteristics is more difficult, especially for unstructured grids and even for structured grids around complex configurations. The unstructured grid generation code VGRID utilizes a variety of sources to provide some control of field grid placement. In particular, the recently developed volume sources have been successfully used to control grid density and orientation in a number of applications. Although this approach can produce very good grids, it can be fairly user-intensive in the placement of the sources, and in the case of grids for sonic boom analysis, the process, including running the grid generation code itself, would have to be repeated for each new Mach number or angle of attack for a configuration.

The SSGRID code avoids these drawbacks by shearing the original grid to align the cells with the free-stream Mach angle and stretching it to reduce the number of points required to reach a given distance from the aircraft. The initial grid is typically generated using a relatively large field grid spacing, except for a region of dense grid underneath the vehicle that is created using a vertical cylindrical volume source. As shearing the grid close to the body might cause it to intersect the wings or other components, an inner cylinder parallel to the body axis and just

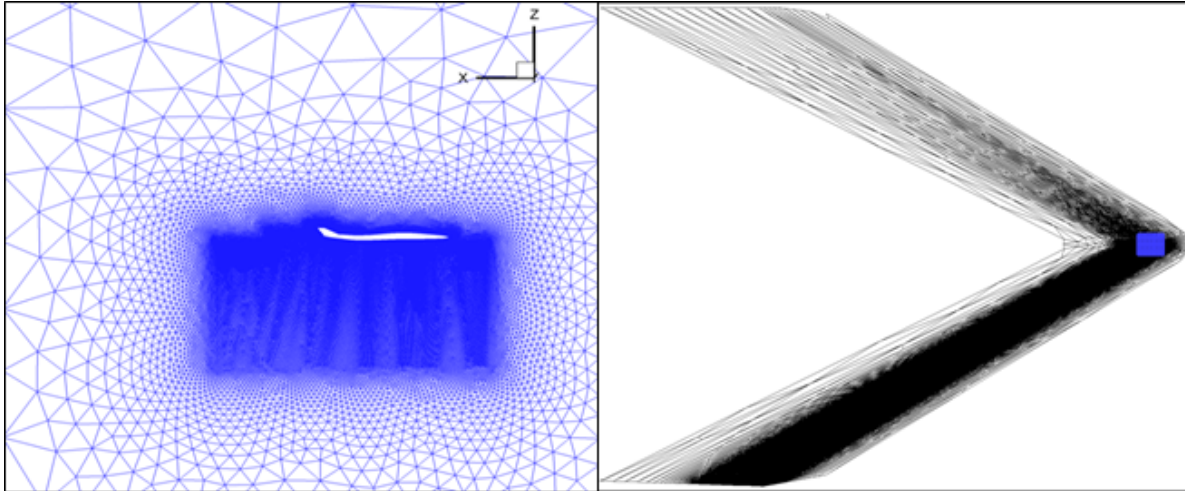
outside the wing tip is defined within which no grid modification occurs. This region tends to be much farther from the keel line of the aircraft than it needs to be and could allow unnecessary dissipation to occur before the sheared grid is reached. To remedy this, a variable inner cylinder radius is used based on the keel line. The keel line, primary inner cylinder radius, and variable radius are all automatically determined in SSGRID based on the aircraft geometry and initial grid characteristics. See Figure 14 for an illustration of grid stretching and shearing.



**Figure 14. Grid stretching and shearing controlled by an inner cylinder.**

The grid points between the inner cylinder and outer boundary are first stretched in radial directions to a user-specified distance and then sheared conically to match the free-stream Mach angle. This aligns the stretch direction with the shocks and expansions. Because the flow gradients are small in the stretch direction, the larger cell size allows these pressure waves to propagate to the mid-field with less dissipation than with the original grid. The grid stretching and shearing in SSGRID will generally produce some negative volume cells, so the grid is run through the ADV grid post-processing code [4] to repair these cells. The entire process is automated and typically takes less than a minute to create a grid for each new pair of a Mach number and an angle-of-attack. The primary user-supplied

inputs are the free-stream Mach number, the angle of attack for the configuration, a reference body length, and the desired location for the stretched outer boundary in body lengths below the aircraft. See Figure 15 for the original VGRID grid and the stretched and sheared grid on the symmetry plane. The highlighted blue region in the stretched and sheared grid was shown in Figure 14.



**Figure 15. Original VGRID grid (left) and stretched and sheared grid (right).**

More details about SSGRID can be found in [4]. In conclusion, the SSGRID approach was by far easier to use, provided more detail in the near field, and retained more of the detail into the mid-field than could be realized from the original VGRID grid. The shearing of the grid to align the faces with the Mach angle provided most of the improvement relative to the original grid, but the grid stretching was also important in allowing the pressure distribution to propagate to the mid-field with minimal dissipation. The use of the new volume sources in VGRID greatly simplified the creation of the dense grid region under the aircraft and kept the overall grid size reasonable. The accuracy of the SSGRID approach apparently is at least comparable to other published adaptation and hybrid grid approaches, with perhaps an advantage in ease of use and improved run time. The ability to quickly develop new grids for different Mach numbers and angles of attack is also a very useful feature of SSGRID.

#### **D. USM3D Engine Simulation**

USM3D provides a variety of engine simulation capabilities, including turbofan and turbojet engine simulations [23][24]. In this paper, only the aft part of engine is modeled, while boundary condition 2 (supersonic outflow) is used for the front cover of the nacelle. This simplification is due to a lack of design capability for a nacelle inlet lip shape and could lead to inaccuracies of the shock at the front of the nacelle. See Figure 4 for the baseline nacelle

geometry. The required USM3D inputs for axisymmetric engine simulation include the six non-dimensional parameters shown in Table 3. These six input parameters are obtained from the NPSS outputs for the engine model that are generated for the baseline configuration. The geometry shape of the aft part of the engine and the engine boundary conditions are correctly modeled by using NPSS outputs and directly affect the plume structure. This allows the current study of how the plume shape affects the mid-field pressure distribution and ground signature.

**Table 3. Nozzle Throat Boundary Conditions**

USM3D non-dimensional parameter definition	USM3D input	Dimensional value
Mass fraction of fuel relative to total exhaust mass flow	0.01729	n/a
Ratio of specific heats	1.322	n/a
Static pressure / $\rho_{\infty} v_{\infty}^2$	7.691	16.45 lb/in <sup>2</sup>
Total pressure / $\rho_{\infty} v_{\infty}^2$	14.258	33.49 lb/in <sup>2</sup>
Gas constant of air / gas constant of jet	0.9994	0.0606 <sup>BTU</sup> / (lbm-° R)
Total temperature / temperature at infinity	4.711	1837 ° R

Cruise Mach = 1.8 at an altitude of 52,000 ft

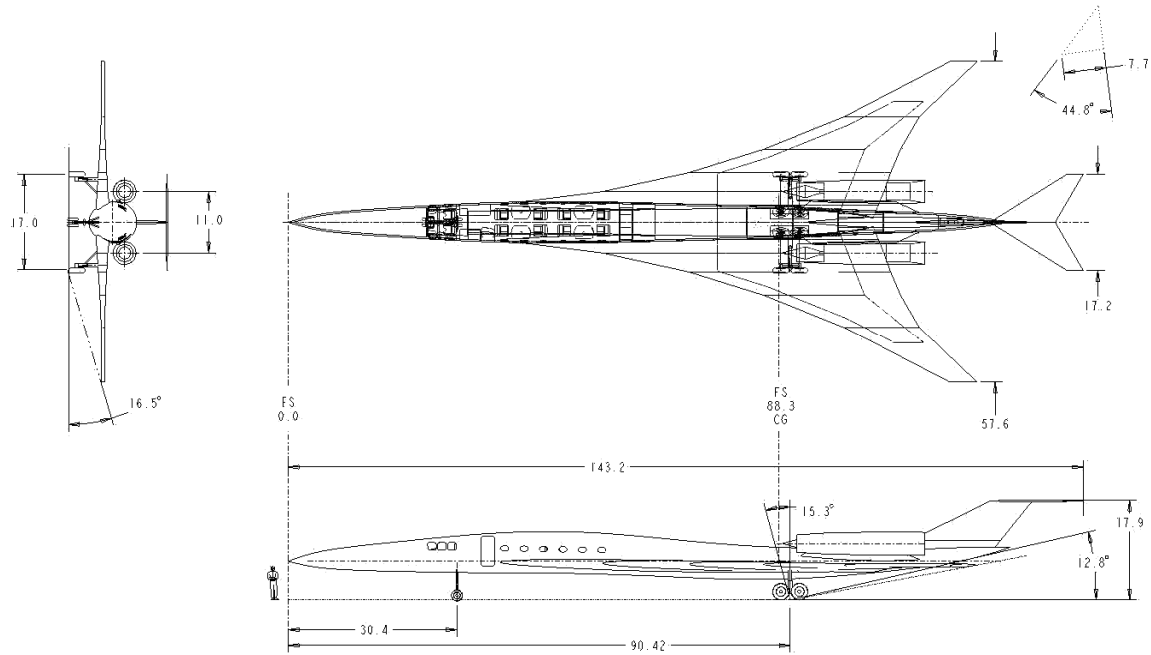
## V. CFD Plume and Boom Analysis Results

Recall that the integrated engine, plume, and boom analysis consists of the following steps: (i) start from a conceptual geometry, (ii) add the nacelle designed by using NPSS, (iii) convert the conceptual geometry to CFD geometry by iPatch, (iv) run AutoSrc, VGRID, and SSGRID for CFD grid generation, (iv) compute the USM3D Euler solution, and (v) use PCBOOM for propagating the mid-field pressure distribution to obtain the sonic boom ground signature of the original concept. In the following subsections, the supersonic concepts developed for this study, computational issues encountered during the CFD analysis, CFD-based boom analysis results, and comparison of low-fidelity and CFD plume shapes are discussed.

### A. Design of the Supersonic Concept for Plume Effect Study

A low-boom configuration with a takeoff gross weight of 100,000 lb was used as a baseline in this study. This configuration was developed based on the premise that a modified ramp signature with an initial overpressure of 0.5 psf would be an acceptable target. See [25] for more details regarding the development of this configuration. The three-view of this configuration is reproduced in Figure 16.



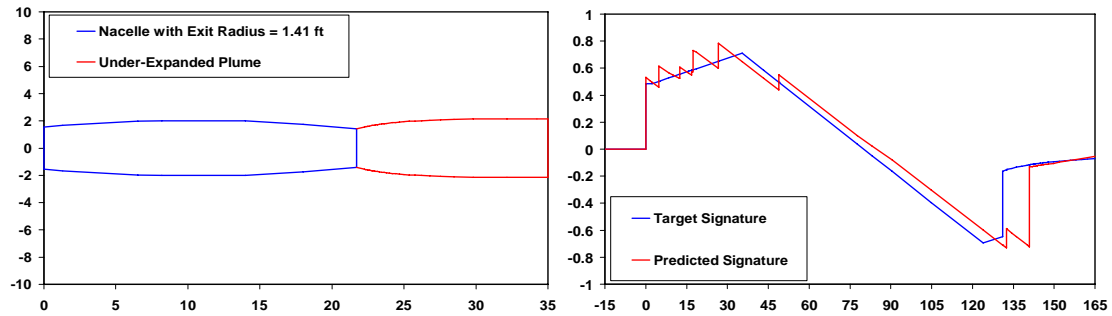


**Figure 16. Three-view of the low-boom baseline configuration.**

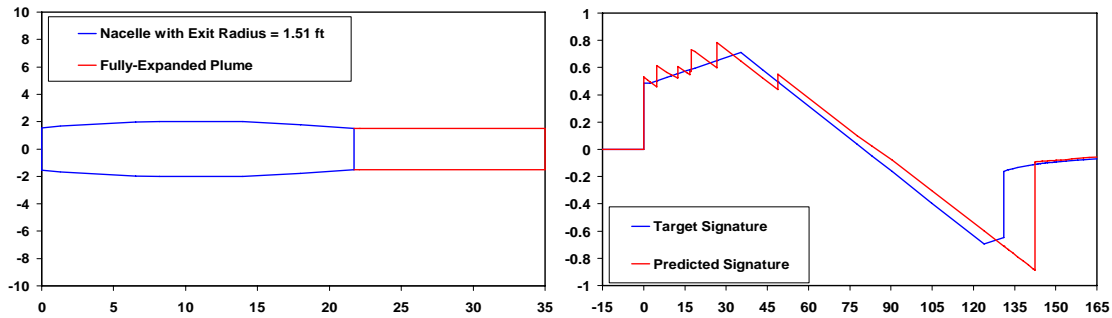
After the original nacelle was replaced with the nacelle generated by using NPSS engine cycle optimization (see subsection II.E) and the fuselage was reshaped for a low-boom signature, the possibility for favorably shaping the aft part of the configuration's ground signature was explored by varying the engine's plume shape. Although this level of detail is usually beyond the ability of the initial conceptual design process, some indication of the impact that the plume shape might have on the ground signature can be calculated by using the low-fidelity plume shape model.

The plume shape was varied by changing the  $K_{noz}$  parameter in the propulsion component of the current ModelCenter process. The  $K_{noz}$  parameter is the ratio of the ideal nozzle area ratio (exit area/throat area) to the actual nozzle area ratio. Varying  $K_{noz}$  leads to changes of the nacelle exit area and plume shape. In general,  $K_{noz} < 1$  or  $K_{noz} > 1$  leads to underexpanded or overexpanded nozzles, respectively. Varying  $K_{noz}$  demonstrated that only underexpanded plumes had any effect on the ground signature. And, because a significant performance penalty is realized for highly over- or underexpanding the nozzle, a plume shape was chosen that was only slightly underexpanded but still had a positive effect on the ground signature as predicted by PBOOM [26] (see Figure 17). For this underexpanded case,  $K_{noz}$  had a value of 1.1. For the second case, a nearly fully expanded plume that looked like a cylinder of the same radius as the nozzle exit was used. Using the low-fidelity plume model, the  $K_{noz}$  parameter was adjusted until a cylindrical plume shape was obtained, which resulted in a  $K_{noz}$  value of 0.96. For the

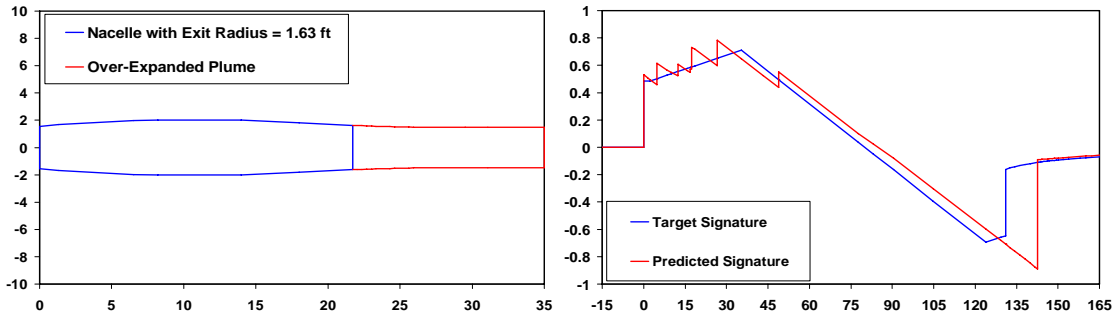
third case, because overexpansion of the nozzle had no detectable effect on the ground signature, a plume that was only slightly overexpanded was selected, which resulted in a  $K_{noz}$  setting of 0.83. The related plume shapes and the ground signatures of the corresponding configurations are shown in Figure 17, Figure 18, and Figure 19. The baseline engine geometry is shown in Figure 4. Note that the only difference in geometry between these nacelles and the baseline nacelle (which is closest to fully expanded as predicted by the low-fidelity plume model) is the nozzle exit radius. Also note that the labels “fully expanded,” “underexpanded” and “overexpanded” are based on the low-fidelity model. Both the low-fidelity model and the CFD calculations show that the *fully* expanded case is actually very slightly overexpanded. Because of the relatively large nozzle internal divergence angle, the low-fidelity model requires a slightly overexpanded nozzle to achieve a cylindrical plume shape.



**Figure 17. Underexpanded nozzle and predicted ground signature by PBOOM.**



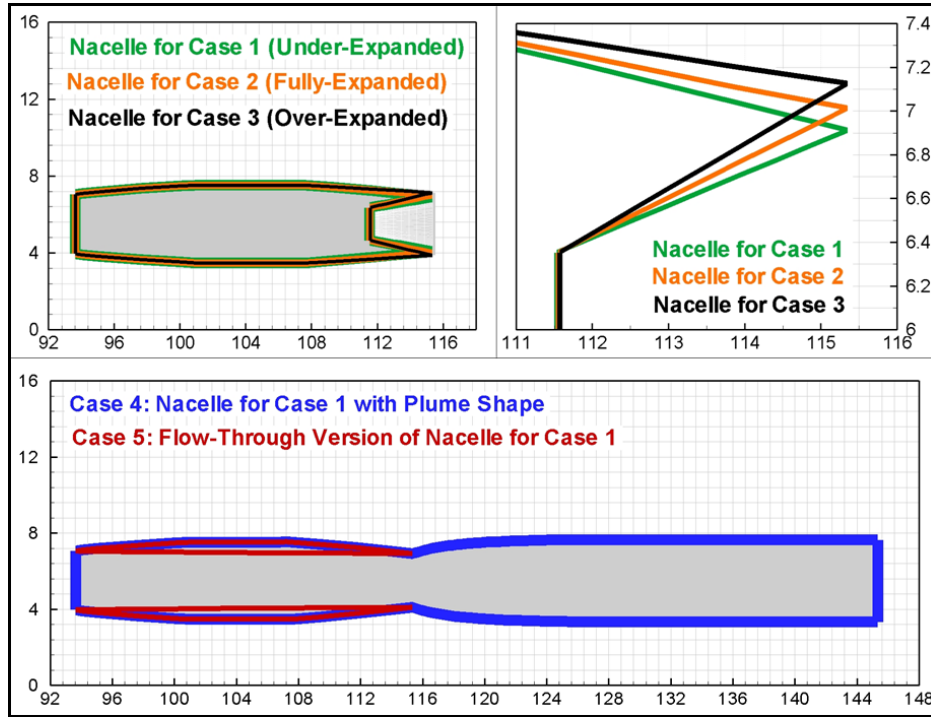
**Figure 18. Fully expanded nozzle and predicted ground signature by PBOOM.**



**Figure 19. Overexpanded nozzle and predicted ground signature by PBOOM.**

Note that a manual redesign of the nacelle OML is usually required to reduce the wave drag of the nacelle. However, the nacelle that is generated by the heuristic process described in subsection II.E is aerodynamically efficient, the propulsion system design parameters are fixed, and the perturbations to the nozzle exit area are small. As a consequence, no redesign of the nacelle OML was required for this study.

The low-fidelity boom analysis treats the nacelle and the plume as a single axisymmetric solid body of revolution. The validity of modeling the plume as a solid body in boom analysis was not verified. To understand the effect of a solid plume shape on the ground signature, the low-fidelity plume shape was attached as an extension of the nozzle as test case 4 for CFD analysis. Finally, as a reference case, the flow-through nacelle of the underexpanded nozzle is included as test case 5. The nacelle shapes for all five cases are shown in Figure 20.



**Figure 20. Nacelle shapes for five test cases.**

## B. Computational Issues

The CFD analysis process was to run AutoSrc, VGRID, SSGRID, and USM3D by using standard scripts with default inputs. The original purpose for asking CFD specialists to run CFD analyses was to fine tune the default inputs and generate intermediate solution files to help debug ModelCenter wrappers for VGRID and USM3D. Unfortunately, after the initial trial runs of two generic configurations to debug the automation process, neither

could be completed on the first try. A VGRID volume grid generation failure and a USM3D convergence failure were encountered during the automated analysis processes.

For several configurations, the grid generation process failed during the development of the surface grid. This problem appeared to be related to the specification of anisotropic stretching for two line sources that were joined but had a significant angle difference, (e.g., at a break in the wing trailing edge). Reducing the number of line sources that were allocated along spanwise lines for lifting-surface components (e.g., a wing) from 50 to 20 or 25 provided a temporary fix for the problem. Future versions of AutoSrc will automatically reduce the anisotropic stretching at corners to conform to the recommendations of the VGRID developer.

Engine simulation and sheared volume grids make the underlying numerical problem very difficult for USM3D to solve, which is quite understandable. In fact, USM3D failed to converge with a Courant Friedrich Levy (CFL) number of 10 for the underexpanded plume, the fully expanded plume, and the solid plume shape cases. The CFL number is used for steady-state calculations to accelerate convergence using non-time-accurate local time stepping. Decreasing the CFL number makes USM3D more robust but slows its convergence. Thus, slow ramp up of the CFL number during the second-order solution phase of USM3D led to converged solutions in these three cases but at the cost of increased computational times. See Table 4 for a summary of the computational costs for all five cases. Total wall time includes the man-hours required for grid generation, grid quality checking, failed trials, and diagnosis of failures. The termination criterion for USM3D execution was based on either a sufficient reduction of the solution residual (i.e., a drop of three orders of magnitude) during the second-order iterations or the difference between the force coefficients of two consecutive runs being close to zero with each USM3D run consisting of 5000 to 8000 iterations. To ensure the numerical convergence at off-body locations, an additional 1000 iterations were run to obtain an additional USM3D solution to ensure that the difference in the  $dp/p$  distributions at six body lengths below the aircraft (i.e.  $R/L=6$ ) for this solution and the last solution was almost zero. The number of iterations used to obtain and verify the USM3D solution is listed as the “actual” number of iterations. The “required” number of iterations in the table is based on the post-analysis (i.e., when USM3D already had a solution that was nearly identical to the final solution). Note that increasing the frequency of checking the termination criterion reduced the difference between the actual and required number of iterations but increased the computational overhead. One important point to note is the added difficulty of coordinating this effort between different disciplines/organizations. The CFD specialists were not working full time on these five test cases, and the submitted CFD runs had to wait in

the NASA supercomputer queues. As a result, the actual turnaround time (i.e., from sending out the input files to receiving the USM3D solution files) was 18 days (including three weekends), or 12 business days. This demonstrates the importance of being able to click a button in ModelCenter to generate CFD solutions.

**Table 4. Computational Costs for Five Cases**

Case	Number of cells (in millions)	Wall time for USM3D (number of CPUs = 96)	Number of iterations		Total wall time
			Actual	Required	
1	22.3	28 h 54 min	37000	26000	34 h
2	22.2	12 h 32 min	18000	10000	35 h
3	22.2	18 h 45 min	26500	18000	24 h
4	20.0	25 h 20 min	46500	42000	31 h
5	13.5	8 h 14 min	18000	15000	13 h

### C. CFD-Based Boom Analysis Results

PCBOOM [7] was used to propagate  $dp/p$  distributions at off-body locations to the ground to predict the sonic boom ground signature. For the Shaped Sonic Boom Demonstrator (SSBD) project, the ground signature prediction by PCBOOM matches the measurement fairly well. The predictions of the initial overpressure and the signature duration are quite accurate, with about a 20 percent overestimate of the maximum and minimum overpressures in the ground signature (Figure. 5b in [8]).

Note that PCBOOM 4.1 has a limitation of 200 on the number of points in the  $dp/p$  distribution. However, PCBOOM 4.1 sometimes yielded erratic solutions when too many points were used as a result of the waveform parameter method used for signature propagation. For example,  $dp/p$  distributions at  $R/L = 5.95$ , 6, and 6.13 are almost identical, but the predicted ground signature at  $R/L = 6$  is quite different from the other two cases if approximately 195  $dp/p$  data points are used as inputs for PCBOOM 4.1 (see Figure 21). Therefore, the original  $dp/p$  distribution was filtered down to about 150 data points for propagation (see Figure 22), which was accomplished by using a piecewise linear approximation algorithm. Then, the predicted ground signature using  $dp/p$  at  $R/L = 6$  was identical to those that were predicted by PCBOOM 4.1 using  $dp/p$  at the other  $R/L$  locations (see Figure 21). In general, the maximum approximation error between the original  $dp/p$  distribution and its piecewise linear approximation was less than 0.00005. Note that an unreleased version of PCBOOM 4.7d does not have this problem and generates the same ground signature using unfiltered  $dp/p$  data.

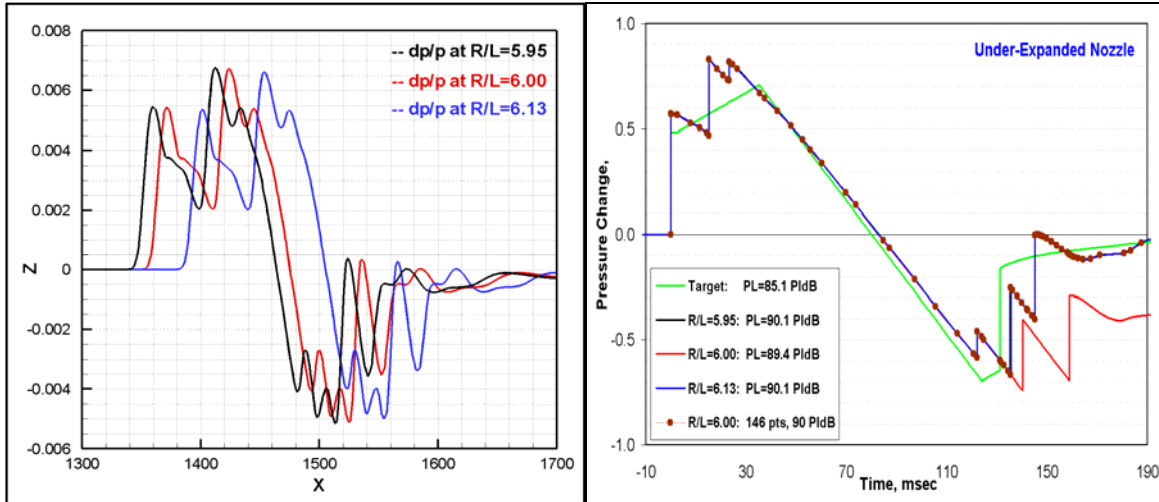


Figure 21. Potential error using waveform parameter method in PCBOOM 4.1.

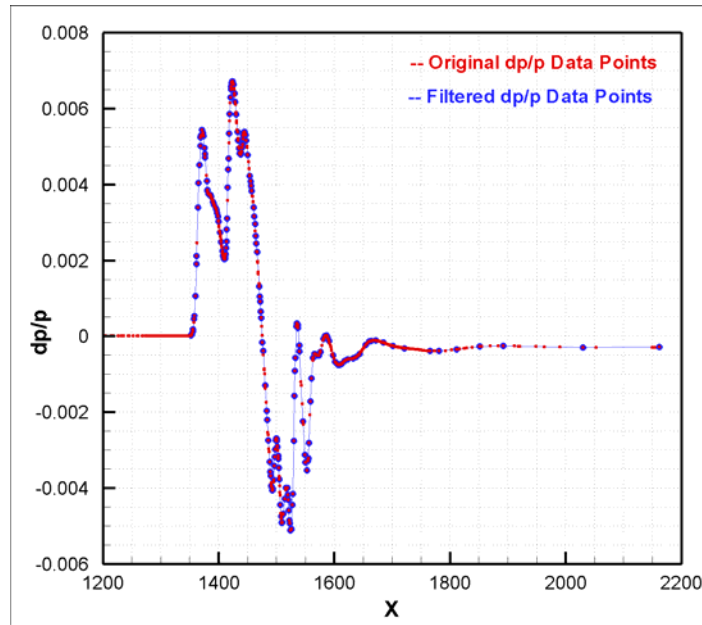


Figure 22. Piecewise linear approximation of  $dp/p$  at  $R/L = 6$  by about 150 points.

The predicted ground signatures show some unexpected results: (i) ground signatures propagated from mid-field pressure distributions are nearly identical to each other no matter which  $R/L$  value is used (see Figure 23), (ii) minor, seemingly random pressure oscillations in the mid-field  $dp/p$  distributions have virtually no effect on the ground signatures (see Figure 24 and Figure 25), and (iii) Euler solutions only show marginal differences in the near- and mid-field pressure distributions for configurations for which the engine plume is modeled or that have flow-through nacelles; the corresponding ground signatures are nearly identical (see Figure 24 and Figure 25).

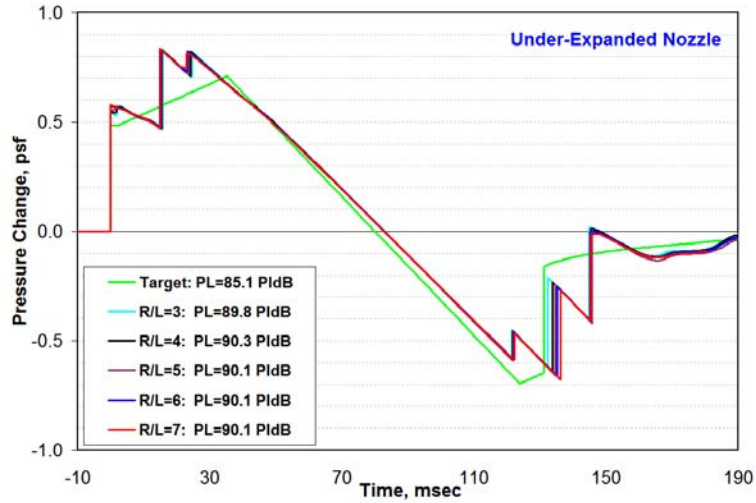


Figure 23. Predicted ground signatures using  $dp/p$  at various locations.

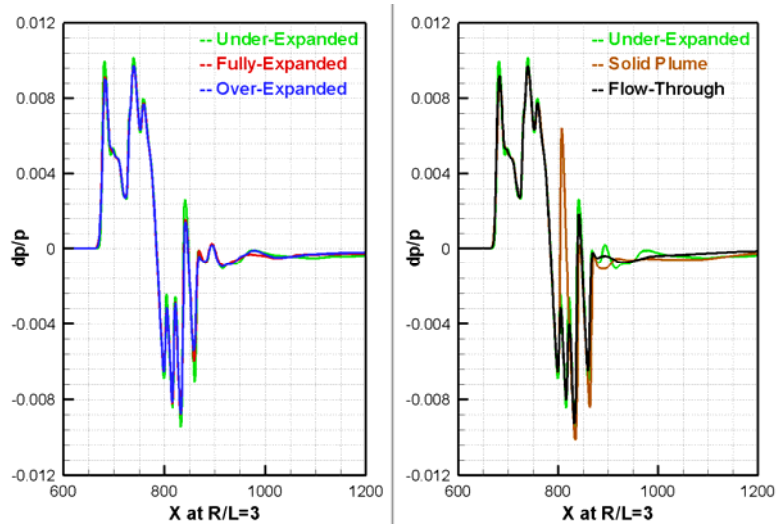


Figure 24. Comparison of  $dp/p$  at  $R/L = 3$  for all cases.

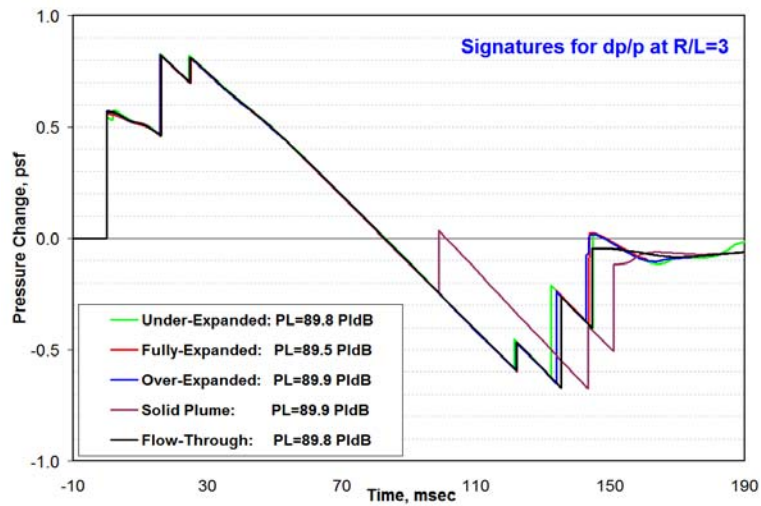


Figure 25. Comparison of ground signatures for all cases.

An Euler solution is not likely to capture the flow physics, including turbulence and possible flow separation, in the plume correctly; furthermore, the accuracy of the off-body pressures is questionable. Therefore, a conclusion that plume shape or engine simulation has no effect on ground signature is still premature. Moreover, all of the plume shapes are nearly fully expanded, and the similarity between the ground signatures for the plume models and the flow-through nacelle might indicate, as expected, that the flow-through nacelle is an excellent approximation of an engine with a fully expanded plume.

However, a surprising result is that the ground signature for the nacelle that was modeled with a solid plume shape is no worse (in terms of the perceived loudness) than the other cases (see Figure 25) because the pressure distribution at  $R/L = 3$  for the nacelle with solid plume shape is so much worse (in terms of the magnitude of  $dp/p$ ) than the other cases (see Figure 24). This is because the spike in the  $dp/p$  shows up well forward of the minimum overpressure; thus, shaping the configuration for a low-boom signature is an intrinsically coupled problem that involves all of the aircraft components and reducing the shock strength locally (from the  $dp/p$  for the solid plume shape to the  $dp/p$  for the flow-through nacelle) might or might not be able to improve the ground signature of the configuration.

#### **D. Comparison of Low-Fidelity and CFD Plume Shapes**

The plume shape can be defined by the boundary between the shear layer of the engine exhaust and the free-stream flow region. The plume boundary might also be interpreted as the velocity stream line starting at the outer diameter of the nozzle exit. The CFD plume visualization can consist of Mach number,  $C_p$  contours, and computed Schlieren plots, which measure the first derivative of the flow density. Thus, defining a definitive plume shape from CFD plume analysis results is difficult.

In this subsection, the low-fidelity plume shape is compared with a Mach contour at the free-stream level ( $M = 1.8$ ) and with velocity stream lines. The low-fidelity predictions of pressure, Mach number, and temperature at the nozzle exit are compared with the average of the CFD results at the nozzle exit and the  $C_p$  contours from CFD results. Also, the computed Schlieren plots are shown for the over- and underexpanded nozzles.

Figure 26 shows the low-fidelity plume shapes for the three cases superimposed over the CFD results of  $C_p$  contours, a contour for  $M = 1.8$ , and the velocity stream lines. The underexpanded case matches perfectly with the



Mach contour at the free-stream level near the nozzle exit, but the plume shape does not follow the velocity stream line.

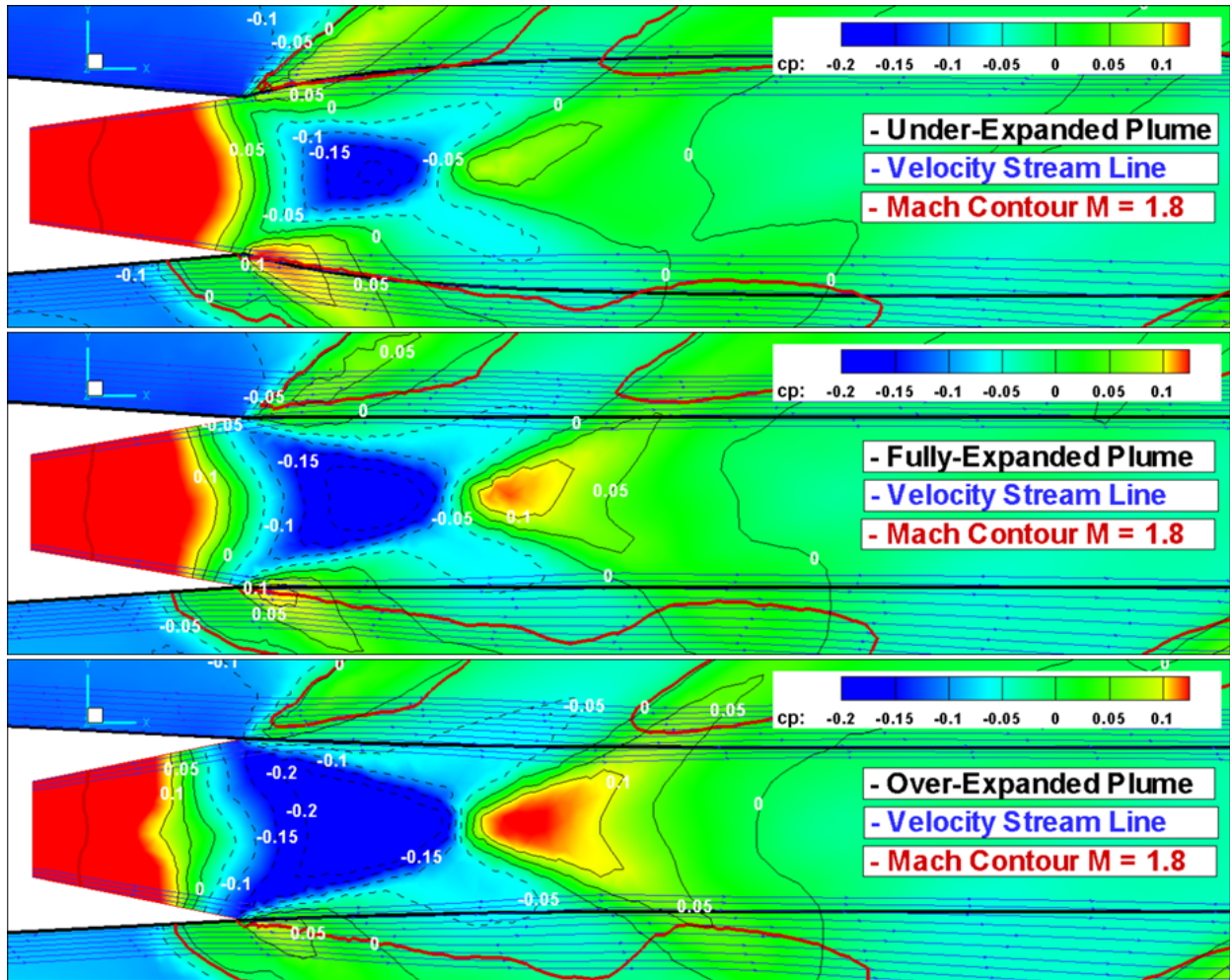


Figure 26. Plume shape comparison including  $C_p$  contours.

For the cases of the fully expanded and the overexpanded nozzles, the low-fidelity plume shapes match the velocity stream lines, with the best match being the slightly overexpanded nozzle. The  $C_p$  contours indicate that both cases have slightly overexpanded plumes with  $P_e/P_\infty < 1$ . In these two cases, the Mach contours at the free-stream level do not correlate with the low-fidelity plume shapes.

Note that classical low-fidelity plume shape prediction methods are typically based on the  $P_e/P_\infty$  value at the nozzle exit. The current low-fidelity method uses the same engine properties at the nozzle throat as the CFD plume model to calculate  $P_e/P_\infty$ . Table 5 shows a comparison of  $C_p$ ,  $M$ , and  $T$  values from NPSS, the present low-fidelity model, and the average of 100 points from the CFD solution at the centerline of the nozzle exit parallel to the XY plane. The  $C_p$  and  $T$  values are reasonable and have expected trends. The Mach numbers at the nozzle exit are not

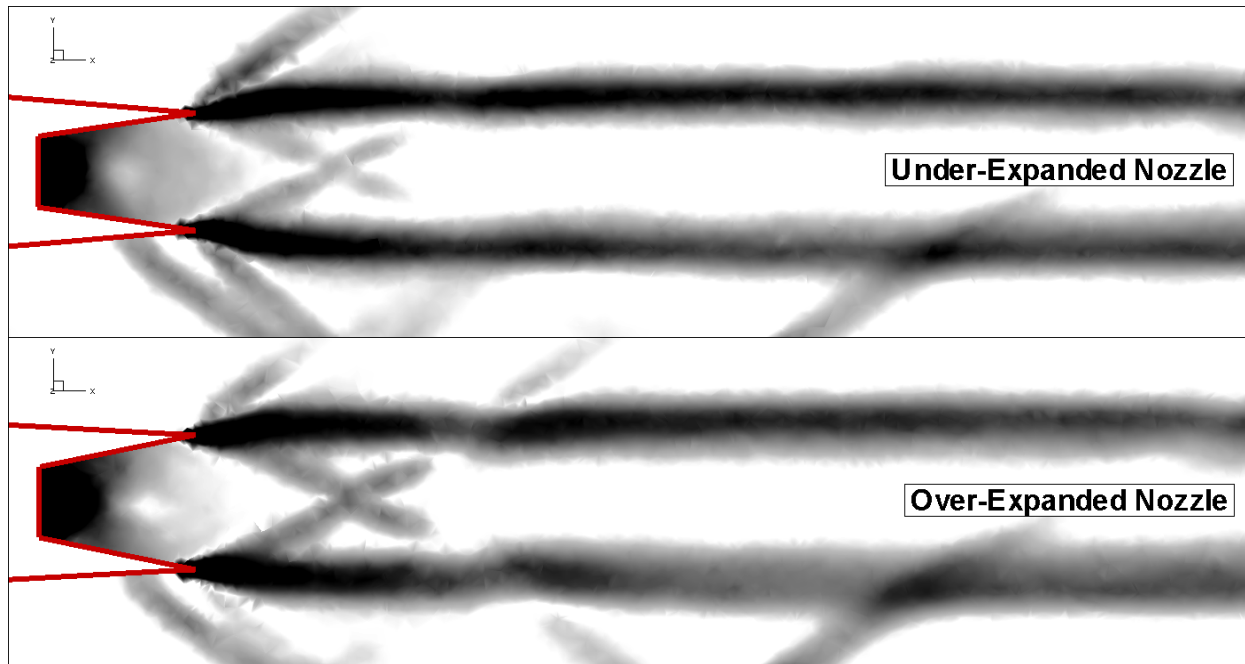
unreasonable but do not show expected trends. For NPSS, the Mach numbers are all increasing, as would be expected for only slightly overexpanded nozzle flow. The low-fidelity plume model uses the axial component of the Mach number, which explains the decreasing Mach number for the overexpanded case relative to the fully expanded case. The CFD Mach number trends, especially from underexpanded to fully expanded cases, are different from either the NPSS or the low-fidelity plume model.

**Table 5. Comparison of Low-Fidelity and CFD Analysis Results at Nozzle Exit**

	Plume	Underexpanded	Fully expanded	Overexpanded
	$NPR_{ideal}$	22.76	21.83	20.82
	NPR	17.59	22.03	27.91
	$K_{noz}$	1.1000	0.9600	0.8300
$C_p$	NPSS	0.0753	-0.0286	-0.1157
	Plume	0.0600	-0.0421	-0.1273
	USM3D	0.0541	-0.0337	-0.1097
$M$	NPSS	2.4773	2.6153	2.7622
	Plume	2.5156	2.6526	2.6223
	USM3D	2.4661	2.3215	2.3645
$T/T_\infty$	NPSS	2.2573	2.1241	1.9915
	Plume	2.1414	2.0044	1.8700
	USM3D	2.1020	1.7188	1.6066
Free-stream conditions		$q = 3.417$ psi		
		$p = 1.528$ psi		
		$T_\infty = 389.97$ °R		
		$a = 968.03$ ft/s		

The similarities in the Mach number contours (see Figure 26) as well as the computed Schlieren plots for the overexpanded and underexpanded cases (see Figure 27) indicate that for these small differences in the degree of nozzle overexpansion or underexpansion, little difference occurs in the plume shape as defined by these criteria.

In summary, for overexpanded nozzles, the low-fidelity plume shape matches the velocity stream line starting at the lip of nozzle exit, while the underexpanded case matches the Mach number contour and computed Schlieren plot from the CFD solutions. Note that the flow physics in the plume region might be incorrect as a result of the inviscid solutions used for engine simulation. Additionally, using low-fidelity models, neither the plume analysis nor the sonic boom analysis are able to capture the effects of any  $dp/p$  distributions that may result from the plume's steady-state shock structure. Further computational and experimental studies are needed to validate the current low-fidelity plume shape analysis and the overall approach to low-fidelity sonic boom analysis with plume effects.



**Figure 27. Computed Schlieren plots of the magnitude of density gradients on XY plane.**

## **VI. Conclusions**

An automated Euler plume and boom analysis process for conceptual aircraft analysis has been successfully demonstrated. With the current man-in-the-loop process for scripted runs of AutoSrc, VGRID, SSGRID, and USM3D, the longest wall time for obtaining a CFD solution was no more than 35 hours (including failed trials). The total turnaround time (i.e., from starting five variations of a low-boom baseline concept in ModelCenter to receiving all the CFD solution files from CFD specialists) was 18 days (including three weekends). The grid generation process has been completely debugged, and automatically generating volume grids for plume and boom analysis of supersonic configurations using some default inputs is feasible. However, no definitive rule exists on how to choose a CFL number for USM3D to obtain a converged CFD solution because of computational difficulty of the CFD plume and boom analysis requirements. A heuristic rule for slow ramp up of the CFL number during the second-order convergence phase of USM3D could be used to control USM3D convergence, but multiple trials of various CFL numbers might be needed to obtain a converged solution. No CFD expertise is required to make trial runs. Moreover, the trouble-free run of the flow-through nacelle case (case 4) confirms that the automated CFD analysis process using VGRID and USM3D is robust and operational if no plume simulation is required.

The five variations of the low-boom baseline differ only in the nacelle: (i) underexpanded nozzle, (ii) fully expanded nozzle, (iii) overexpanded nozzle, (iv) nacelle extended by the underexpanded plume shape, and (v) the flow-through version of the underexpanded nozzle. Here, the classification of nozzles is based on a low-fidelity plume shape prediction model that is introduced in this paper, and the nacelle geometry is based on the NPSS outputs. The low-fidelity plume shape prediction model and the USM3D engine simulation use the same set of engine parameters from the NPSS outputs. For the underexpanded nozzle, the Mach number contour at free-stream level matches well with the low-fidelity plume shape. For the overexpanded nozzle, the CFD velocity stream line beginning at the lip of the nozzle exit is in good agreement with the low-fidelity plume shape. However, the best way to define a plume shape from the CFD solution is still unknown when the engine is a part of a complete aircraft and when the objective is to obtain a low-boom ground signature.

The PCBOOM prediction of the ground signature using the  $dp/p$  distribution in the mid-field (3 to 10 body lengths below the aircraft) is quite robust with respect to  $R/L$  perturbations. In fact, the predicted ground signatures are nearly identical to each other as long as the  $dp/p$  distribution is sampled at more than three body lengths below the configuration.

The low-fidelity boom analysis indicates that a benefit could be realized for using an underexpanded plume to shape the aft signature. PCBOOM propagation of  $dp/p$  distributions shows that the ground signatures of the variations of the low-boom baseline, except the solid plume shape case, are nearly identical (i.e., the CFD plume shapes and engine simulation have no significant effect on ground signature in this study). Because all of the CFD plumes are close to fully expanded, these results may indicate that a configuration with an almost fully expanded plume has a similar  $dp/p$  distribution in the mid-field to the same configuration with a flow-through nacelle. Thus, for CFD boom analysis of such a configuration, a plume simulation may not be necessary in the CFD analysis. This result is actually a more desirable outcome for the practical development of a low-boom supersonic aircraft because no one wants to hear a significant increase in loudness of sonic boom when the engine performance deviates slightly from the optimal operating conditions during cruise. Because the plume and sonic boom analyses cannot capture the effects of any  $dp/p$  distributions that may result from the plume's internal steady-state shock structure using the low-fidelity models, the only effect that might be captured is the initial shock that is caused by interaction of the free-stream with the plume.

With the plume modeled as a solid body based on the low-fidelity shape (i.e., the solid plume shape case), a spike appears much further forward in the signature than the spike in the low-fidelity analysis. This result does suggest that the underexpanded nozzles may have some effect on the ground signature if the plume is sufficiently underexpanded, so that its effect on the mid-field pressure distribution is more like that of the solid plume shape. This result also suggests that the low-fidelity plume shape and/or sonic boom analysis is overpredicting the effect of the plume on the ground signature. Finally, an Euler solution is unlikely to be able to capture the flow physics in the plume correctly, and the weak shocks caused by the plume may not be propagated correctly to the mid-field without sufficient grid resolution below the plume region. Therefore, a decision as to whether the plume shape has an appreciable effect on ground signature is still premature.

The integrated engine, plume, and CFD analyses demonstrate the need for further improvement of (i) NPSS, for more detailed supersonic inlet and nacelle geometry definition; (ii) VGRID, for robustness with respect to user source inputs; and (iii) USM3D, for easy-to-use convergence control. Moreover, the inconsistency between low-fidelity and CFD plume and boom analysis results calls for further computational and experimental studies to improve and validate the current variable fidelity approach for boom analysis and boom mitigation of supersonic concepts with plume effects.

## Appendix: Sample Topology Specification File for iPatch

Operation\_Mode: Grouping  
Group\_Name: Fuselage  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 1  
Operation\_Mode: Grouping  
Group\_Name: Wing  
Total\_Number\_of\_Grids: 6  
Grids\_Indices: 2 3 4 5 6 7  
Operation\_Mode: Grouping  
Group\_Name: WingTip  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 8  
Operation\_Mode: Grouping  
Group\_Name: NacOut  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 9  
Operation\_Mode: Grouping  
Group\_Name: NacFrt  
Total\_Number\_of\_Grids: 2  
Grids\_Indices: 10 11  
Operation\_Mode: Grouping  
Group\_Name: NacAft  
Total\_Number\_of\_Grids: 2  
Grids\_Indices: 12 13  
Operation\_Mode: Grouping  
Group\_Name: NacIn  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 14  
Operation\_Mode: Grouping  
Group\_Name: Pylon  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 15  
Operation\_Mode: Grouping  
Group\_Name: Vtail  
Total\_Number\_of\_Grids: 2  
Grids\_Indices: 16 17  
Operation\_Mode: Grouping  
Group\_Name: Htail  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 18  
Operation\_Mode: Grouping  
Group\_Name: HtailTip  
Total\_Number\_of\_Grids: 1  
Grids\_Indices: 19  
  
Operation\_Mode: Intersection  
1st\_Grid\_Intersecting\_2nd\_Grid: 2 1  
Intersection\_Index: 2  
Operation\_Mode: Intersection  
1st\_Grid\_Intersecting\_2nd\_Grid: 15 1  
Intersection\_Index: 2  
Operation\_Mode: Intersection  
1st\_Grid\_Intersecting\_2nd\_Grid: 16 1  
Intersection\_Index: 2  
Operation\_Mode: Intersection  
1st\_Grid\_Intersecting\_2nd\_Grid: 17 18  
Intersection\_Index: 2  
  
Operation\_Mode: End

## References

- [1] Lytle, J., Follen, G., Naiman, C., Evans, A., Veres, J., Owen, K., and Lopez, I., "Numerical Propulsion System Simulation (NPSS) 1999 Industry Review," NASA-TM-209795, August 2000.
- [2] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.
- [3] VGRID Unstructured Grid Generator, <http://tetruss.larc.nasa.gov/vgrid/>, accessed on August 18, 2008.
- [4] Campbell, L., Carter, M., and Deere, K., and Waithe, K., "Efficient Unstructured Grid Adaptation Methods for Sonic Boom Prediction," AIAA Paper 2008–7327, August 2008.
- [5] Frink, N., Pirzadeh, S., Parikh, P., Pandya, M., and Bhat, M., "The NASA Tetrahedral Unstructured Software System," *The Aeronautical Journal*, Vol. 104, No. 1040, 2000, pp. 491–499.
- [6] USM3D Users Manual Version 6.0, <http://tetruss.larc.nasa.gov/usm3d/>, accessed on August 18, 2008.
- [7] Plotkin, K., and Grandi, F., "Computer Models for Sonic Boom Analysis: PCBoom4, CABoom, BooMap, CORBoom," Wyle Report WR 02–11, June 2002.
- [8] Morgenstern, J., Arslan, A., Lyman, V., and Vadyak, J., "F-5 Shaped Sonic Boom Demonstrator's Persistence of Boom Shaping Reduction through Turbulence," AIAA Paper 2005–0012, January 2005.
- [9] Haering, E., Murray, J., Purifoy, D., Graham, D., Meredith, K., Ashburn, C., and Stucky, M., "Airborne Shaped Sonic Boom Demonstration Pressure Measurements with Computational Fluid Dynamics Comparisons," AIAA Paper 2005–0009, January 2005.
- [10] Ozcer, I., and Kandil, O., "FUN3D/OptiGRID Coupling for Unstructured Grid Adaptation for Sonic Boom Problems," AIAA Paper 2008–61, January 2008.
- [11] Howe, D., Waithe, K., and Haering, E., "Quiet Spike Near-Field Flight Test Pressure Measurements with Computational Fluid Dynamics Comparisons," AIAA Paper 2008–128, January 2008.
- [12] Lamar, J., and Abdol-Hamid, K., "USM3D Unstructured Grid Solutions for CAWAPI at NASA LaRC," AIAA Paper 2007–682, January 2007.
- [13] Elmiligui, A., Abdol-Hamid, K., and Massey, S., "PAB3D Simulations for the CAWAPI F-16XL," AIAA Paper 2007–491, January 2007.
- [14] Ball, W., and Hickcox, T., "Rapid Evaluation of Propulsion System Effects," Technical Report AFFDL–TR–78–91, Volume I–Final Report, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, July 1978.
- [15] Onat, E., and Klees, G., "A Method to Estimate Weight and Dimensions of Large and Small Gas Turbine Engines," NASA–CR–159481, January 1979.
- [16] Barger, R., and Melson, N., "Comparison of Jet Plume Shape Predictions and Plume Influence on Sonic Boom Signature," NASA–TP–3172, March 1992.
- [17] Nash, K., Whitaker, K., and Freeman, M., "Predicting Exhaust Plume Boundaries with Supersonic External Flows," *Journal of Spacecraft and Rockets*, Vol. 31, No. 5, 1994, pp. 773–777.
- [18] AMES Research Staff, "Equations, Tables, and Charts for Compressible Flow," NACA Report 1135, 1953.
- [19] Walatka, P., Buning, P., Pierce, L., and Elson, P., "PLOT3D User's Guide," NASA–TM–101067, March 1990.
- [20] Samareh, J., "GridTool: A Surface Modeling and Grid Generation Tool," Proceedings of the Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions, NASA–CP–3291, May 1995.
- [21] GridTool, <http://geolab.larc.nasa.gov/GridTool/>, accessed on August 18, 2008.
- [22] Pirzadeh, S., "Advanced Unstructured Grid Generation for Complex Aerodynamics Applications," AIAA Paper 2008–7178, August 2008.
- [23] Abdol-Hamid, K., Frink, N., Deere, K., and Pandya, M., "Propulsion Simulations Using Advanced Turbulence Models With the Unstructured-Grid CFD Tool, TetrUSS," AIAA Paper 2004–714, January 2004.
- [24] Boundary Conditions for USM3Dns, <http://tetruss.larc.nasa.gov/usm3d/bc.html>, accessed on August 18, 2008.
- [25] Li, W., Shields, E., and Le, D., "Interactive Inverse Design Optimization of Fuselage Shape for Low-Boom Supersonic Concepts," *Journal of Aircraft*, Vol. 45, No. 4, 2008, pp. 1381–1398.
- [26] Coen, P., "Development of a Computer Technique for the Prediction of Transport Aircraft Flight Profile Sonic Boom Signatures," NASA–CR–188117, March 1991.