NASA/TM–2010-216711

Initial Multidisciplinary Design and Analysis Framework

L. P. Ozoroski, K. A. Geiselhart, S. L. Padula, W. Li, E. D. Olson, and R. L. Campbell
Langley Research Center, Hampton, Virginia

E. W. Shields
ATK, Langley Research Center, Hampton, Virginia

J. J. Berton, J. S. Gray, S. M. Jones, C. G. Naiman, and J. A. Seidel
Glenn Research Center, Cleveland, Ohio

K. T. Moore and B. A. Naylor
DB Consulting Group, Inc., Glenn Research Center, Cleveland, Ohio

S. Townsend
ASRC Aerospace Corporation, Glenn Research Center, Cleveland, Ohio

June 2010

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at 443-757-5803

- Phone the NASA STI Help Desk at 443-757-5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

Initial Multidisciplinary Design and Analysis Framework

L. P. Ozoroski, K. A. Geiselhart, S. L. Padula, W. Li, E. D. Olson, and R. L. Campbell
Langley Research Center, Hampton, Virginia

E. W. Shields
ATK, Langley Research Center, Hampton, Virginia

J. J. Berton, J. S. Gray, S. M. Jones, C. G. Naiman, and J. A. Seidel
Glenn Research Center, Cleveland, Ohio

K. T. Moore and B. A. Naylor
DB Consulting Group, Inc., Glenn Research Center, Cleveland, Ohio

S. Townsend
ASRC Aerospace Corporation, Glenn Research Center, Cleveland, Ohio

Table of Contents

Abstract

Within the Supersonics (SUP) Project of the Fundamental Aeronautics Program (FAP), an initial multidisciplinary design & analysis framework has been developed. A set of low- and intermediate-fidelity discipline design and analysis codes were integrated within a multidisciplinary design and analysis framework and demonstrated on two challenging test cases. The first test case demonstrates an initial capability to design for low boom and performance. The second test case demonstrates rapid assessment of a well-characterized design. The current system has been shown to greatly increase the design and analysis speed and capability, and many future areas for development were identified. This work has established a state-of-the-art capability for immediate use by supersonic concept designers and systems analysts at NASA, while also providing a strong base to build upon for future releases as more multifidelity capabilities are developed and integrated.

2 Introduction

2.1 Background

The primary challenge in the design of a practical supersonic cruise vehicle is to increase the efficiency and to remove the environmental and performance barriers. Recognizing that these barriers are not captured by traditional disciplinary analysis, the top-level goal of the Supersonic Project (SUP) is to develop rapid multidisciplinary analysis and optimization (MDAO) methods to address this technical challenge. The highly integrated design of a supersonic vehicle requires the simultaneous optimization of airframe, inlet, engine, and nozzle characteristics for high efficiency within sonic boom and community noise constraints. Moreover, the requirement for highly integrated designs necessitates the modeling of the interactions of the airframe and propulsion system earlier in the design process. Typical design optimization problems in supersonics require a balance between the long slender shape that is needed for a feasible low sonic boom design and the higher wing aspect ratio and large noise suppression nozzles that are needed to comply with community noise constraints. To produce a commercially viable vehicle, all of these design drivers must be considered while optimizing cruise efficiency. Understanding and exploiting the interactions of these supersonic technologies is the key to the creation of practical designs. Thus, conceptual design of efficient and environmentally compatible supersonic aircraft requires integrated systems analysis, optimization, and visualization tools.

To address this need, NASA has undertaken the development of an MDAO capability. As stated in the "Vision and Scope Document for the Fundamental Aeronautics Program Multidisciplinary Analysis and Optimization Capability**:"**

> *The vision of the MDAO capability is to facilitate a seamless transition between single-discipline and multidisciplinary analyses by providing systematic processes and an intelligent computational environment for managing multidisciplinary variable-fidelity tools that enable system analysis and optimization at primarily the conceptual and preliminary design stages for all flight regimes of conventional and unconventional vehicle classes. The MDAO capability will be compatible with state-of-the-art computing*

architectures, comply with established standards, support distributed heterogeneous platforms, and reduce overall setup and run time.

The SUP and Subsonic Fixed Wing (SFW) Projects within the FAP have collaborated on this effort, and while each project has their own project milestones, many of the tools and subprocesses within the MDAO framework capability are crosscutting. The focus of this report will only be on the tools and processes being used and developed for use within the SUP Project.

The initial system (referred to as Gen 0) that is documented herein is just the first step to the envisioned long-term goal of having an integrated set of tools and processes from empirical preconceptual design tools to high-fidelity design and analysis tools, which are capable of assessing and designing highly integrated supersonic vehicles. Many groups in both academia and industry have or are currently developing similar systems for their internal design and analysis work. Our goal is to develop a state-of-the-art MDAO capability that can be used for both internal design development and systems analysis studies of internal and external concepts, as well as provide nonproprietary modules for use in academia and industry where similar capabilities are needed. An additional goal is to identify the major gaps in analysis capabilities that are required for efficient MDAO.

Until this effort was undertaken, the aircraft sizing and synthesis code FLOPS (ref. 1) had long been NASA's primary tool for aircraft conceptual design, analysis, and optimization. FLOPS is a monolithic multidisciplinary system for conceptual and preliminary design and evaluation of advanced aircraft concepts. It consists of nine primary modules: 1) weights, 2) aerodynamics, 3) engine cycle analysis, 4) propulsion data scaling and interpolation, 5) mission performance, 6) takeoff and landing, 7) noise footprint, 8) cost analysis, and 9) program control. Given mission requirements and related inputs, FLOPS is capable of sizing an aircraft and optimizing its performance subject to a variety of constraints. Optimization of the configuration can be accomplished through the use of internal FLOPS optimization capabilities.

For subsonic concepts, FLOPS has a long history and, other than for the propulsion system, is capable of the complete analysis from weights to aerodynamics to mission performance. The monolithic FLOPS code is quite reliable and computationally more efficient when the individual discipline analyses are very fast and the study concept is considered conventional relative to the internal database and scaling laws that are inherent to the code. For supersonic concepts, however, many internal modules in the code historically have not been used but have been supplemented with data from other analysis tools. These external analysis codes were run manually and required the development of scripts for data manipulation when working with FLOPS. While this provides some ability to input external data (of whatever fidelity is available) into FLOPS, it is not particularly convenient. Additionally, many of the other disciplines included in FLOPS are based on data and experience for subsonic aircraft and, therefore, are not adequate for supersonic aircraft.

The current implementation of the supersonics MDAO capability is built around using FLOPS for the initial first cuts at high-order system-level metrics but then taking advantage of the external analysis codes for supplying input data to the aircraft weight, mission performance, and takeoff and landing analysis modules within FLOPS, post processing with additional analysis codes, and moving the majority of the optimization work to an external optimizer. In this implementation, FLOPS is used only to execute the mission performance optimization and the takeoff analysis. Adoption of a software integration framework allows the various discipline modules to be updated outside the core synthesis module and easily replaced. In addition, the mission analysis and weights modules are independent. This implementation leads to a much more flexible MDAO environment.

When higher fidelity design and analysis tools are integrated into a convenient software integration framework, designers and systems analysts can complete design cycles and trade studies faster and with improved confidence. However, an integrated MDAO framework capability has benefits beyond design speed and confidence. A powerful

software framework can include a generic set of tools, such as design of experiments (DOE), guided search, optimization, probabilistic analysis, response-surface methods, and geometry visualization. A systems analyst can learn to use these generic tools quickly and then employ the same tools on every subsequent project. Moreover, the framework allows an existing configuration to be quickly reconfigured. Thus, the supersonic model that was created for the MDAO milestone provides a point of departure for future design studies. In addition, each completed study creates one or more specific models that simultaneously archive past results and enable new studies. Conceptual design tools that are integrated into a well-maintained software framework allow systems analysts to perform and document new studies without tedious data entry or an endless learning curve.

Some of the goals of the MDAO capability presented here are similar to those of a previous effort known as the Conceptual Design Shop (CDS). The CDS project ended prematurely after completing just two years of a planned six-year effort, so its goals, which can be viewed as a subset of the present proposed MDAO capability, were not realized. However, the CDS effort did produce a prototype system that is used currently to support systems analysis. This system used a commercially available software framework to connect a variety of low-fidelity analysis codes and to enable system-level MDAO capabilities. The work started by the CDS team has proven to be a valuable stepping stone toward the achievement of the Gen 0 goals.

2.2 Milestone and Metrics

The MDAO milestone metric reads as follows:

"Demonstrate a two day analysis cycle time for integrated prediction of cruise efficiency and environmental compatibility given a vehicle concept."

To achieve this milestone, many different discipline analysis codes of varying levels of fidelity need to be integrated into an MDAO framework. Several of these codes required some new development, and one process (plume prediction) was newly developed to fill a gap for that discipline at the low-fidelity level. The list of required capabilities and a discussion of the development of the overall integrated framework are included in section 3. The development and integration of the individual analysis codes is discussed in section 4. To verify that the discipline analysis codes were interacting properly, a supersonic business jet was designed. The use of the MDAO framework for this typical design scenario is documented in section 5.1. The two-day assessment of a vehicle concept is discussed in section 5.2, and an overall assessment of the software framework is given in section 5.3

3 Required Capabilities

The Gen 0 milestone mandates an integration of low- and intermediate-fidelity analysis codes into an MDAO framework. To build upon past and concurrent work, the ModelCenter® framework, a product of Phoenix Integration, Inc. (ref. 2),[1] was chosen for this effort. The ModelCenter framework was one of several frameworks that were evaluated during the CDS project. Thus, the use of the ModelCenter framework leverages existing expertise in building and executing process models; further, a number of analysis codes that were wrapped for execution in a ModelCenter process model were available. In addition, experimentation with the CDS prototype revealed some best practices for creating models and some framework features that could be exploited when creating new components. Additionally, ModelCenter has proven to be a solid framework, providing several

---

[1] The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

different ways to wrap a code, as well as a useful set of built-in tools. The discipline analysis models and concept studies presented in sections 4 and 5 highlight the capabilities of the ModelCenter-based MDAO framework.

The analysis capabilities that were identified at the beginning of this effort for inclusion in the Gen 0 capability for supersonic design and assessment are summarized in table 1. The analysis codes which currently enable these capabilities are listed in table 2. Several of the codes in Table 2 required development, modification, and/or integration during this reporting period. A brief description of those codes and the technical challenge that was involved with integration into the MDAO framework are discussed in section 4. While all of these codes have been tested in the Gen 0 framework (either as a standalone component or as a fully integrated analysis process in ModelCenter), the overall robustness of the process has yet to be fully evaluated.

Table 1. Required Capabilities for Initial Framework

Geometry
Parametric capability
Watertight for computational fluid dynamics (CFD)
Propulsion System
Engine-cycle analysis
Engine sizing, weight, and flow path
Shape Design
Wing and fuselage aerodynamic design
Shaping for low boom
Aerodynamics
High-speed aerodynamic performance
Low-speed aerodynamic performance
Engine plume prediction
Weights
Empirical subsystem
Semi-empirical primary structure
Mission Analysis
Climb and cruise performance
Fuel use and emissions estimates
Takeoff and landing performance and flight paths
Human Acceptance
Sonic boom
Community noise
Cost
Stability and control

Table 2. Technical Challenges to Meet Initial Framework Milestone

Code name	Description	Technical challenge	Status
NPSS	Numerical propulsion system simulation	Create plug-in	Functional
WATE++	Weight analysis of turbine engines	Create plug-in	Functional
VSP	Vehicle sketch pad	Improve import/export capabilities	Fully functional
IPATCH	Create watertight	Combine components to	Functional

	geometry	form outer mold line	
AutoScr	Automatic source and line placement for input to CFD solver	Capture knowledge from CFD experts	Functional
WDES	Low-fidelity supersonic wing design and performance	Previously integrated	Fully functional
AWAVE	Low-fidelity wave drag prediction	Previously integrated	Fully functional
CDF	Low-fidelity skin friction prediction	Previously integrated	Fully functional
AERO2S	Low-fidelity low-speed aero performance	Automate reconfigurable flap definition	Fully functional
PBOOM	Low-fidelity sonic boom prediction	Previously integrated	Fully functional
BOSS	Boom optimization using smoothest shape modification	Incorporate suggestions from users	Fully functional
HYBRID	Sonic boom minimization	Previously integrated	Fully functional
PDCYL	Point design of cylindrical bodies	Adapt to supersonics	Needs validation for supersonics
MaSCoT	Matlab Stability and Control (S&C) Toolbox	Links to geometry, weights, and aerodynamics	Needs validation for supersonics
VORVIEW (VORLAX)	Low-fidelity stability and control derivative prediction	Previously integrated	Functional but not robust
FLOPS	Flight optimization	Create separate mission and weights components and improve usability	Fully functional
ALCCA	Aircraft life cycle cost analysis	Produce credible cost estimates	Fully functional
ANOPP	Aircraft noise prediction program	Links to mission analysis and propulsion design	Jet and fan noise sources tested
VGRID	Unstructured grid generation	Trade-off between accuracy and computer time	Not robust, still in debug mode
USM3D	Euler/Navier-Stokes solver for unstructured, tetrahedral meshes	Automate input generation	Not robust, still in debug mode
PCBOOM	Sonic boom analysis	Generate input from mid-field pressure distribution	Functional
Loudness	Updated version of Mark VII code	Automate evaluation of boom signature	Fully functional

4 MDAO Framework Status
4.1 Overall Process Model

A top-level view of the current overall Supersonics Design Analysis and Optimization process is shown in figure 1. Each oval icon is an assembly and represents a subprocess with one or more analysis components.
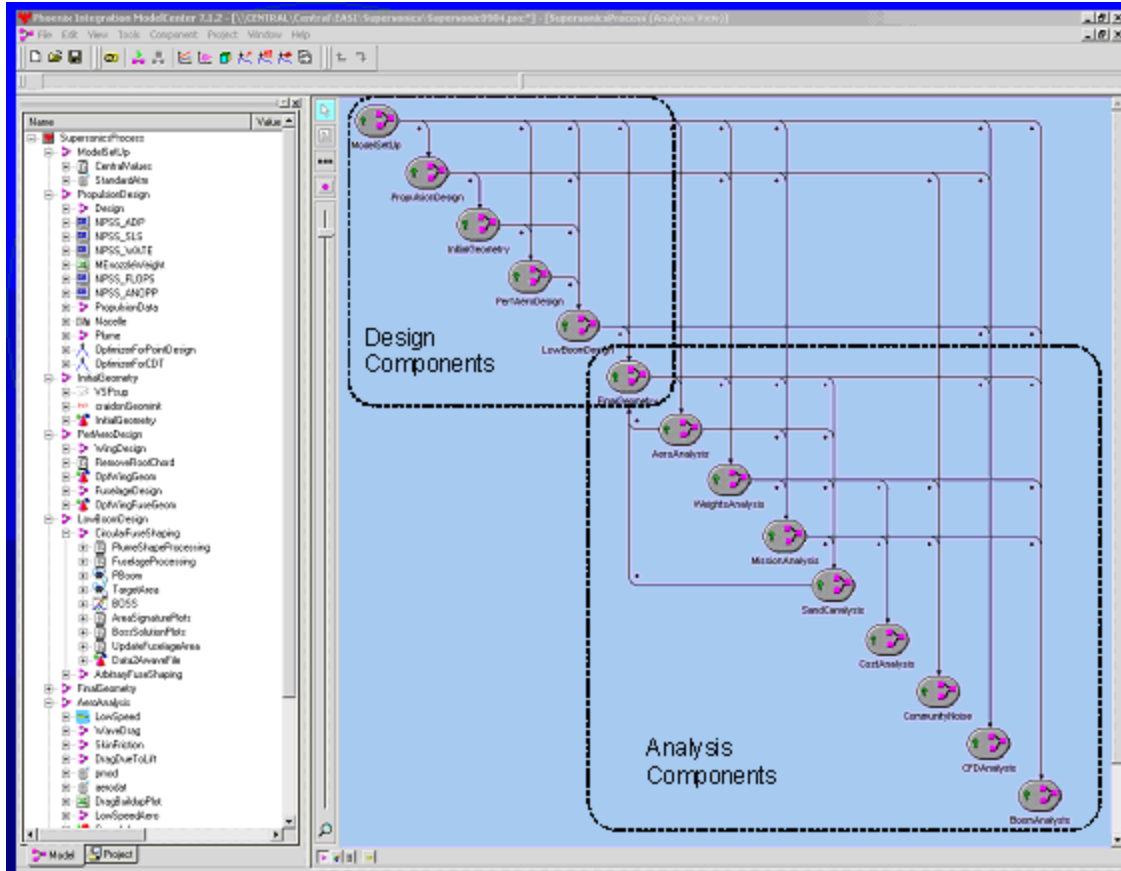

Figure 1. Top-level view of the supersonics framework.

The two blocks that are highlighted by the dotted lines indicate assemblies that are used primarily for design and assemblies that are used primarily for analysis. The upper set of assemblies contains initial concept design components that work best with a concept designer "in the loop." These components provide graphical inspections and better control over certain aspects of the design process. The lower set of assemblies contains components that work well in automated mode. These analysis components are designed to run DOE or trade studies as permutations from a baseline design. The current SUP MDAO Framework contains 215 components and approximately 272,000 data variables. It incorporates commercial software through plug-ins, provides geometry visualization, has components distributed across various computer platforms (i.e., computers with Microsoft® Windows®, Silicon Graphics® Unix® and Red Hat® Linux operating systems) and has components written in Sun Microsystems® Java®, FORTRAN, C, C++, Microsoft VBS®, and other computer languages.

The entire design and analysis process can be controlled with a system-level optimizer. To run an optimization from a defined initial concept, the suggested approach is to turn off or remove some of the design components and primarily use the analysis components. Otherwise, the optimization process must execute the design components with the user-interactive features turned off. Thus, many components will execute even though each one is

producing unchanged information. This situation is obviously undesirable because the optimization process is spending execution time needlessly.

An improved system-level optimization process requires logic control to skip unneeded assemblies and components. At the current time, Phoenix Integration is developing a flexible capability for adding logic nodes to processes in ModelCenter. The Phoenix Integration effort to develop process control will complement the MDAO framework effort to develop the multifidelity components and subprocesses that are required for supersonic applications. While logic control is possible in the current system, implementation can be difficult and is not intuitive. The decision was made to delay incorporating process logic flow until the future capability is available, which will provide a much faster method for adding and controlling process flow in a multifidelity system.

Figure 1 provides an overview of the supersonics process model. The following sections of this report break down the various assemblies shown in the overall process model and discuss the development, usage, and capabilities of each new component.

4.2   Propulsion

The propulsion system design assembly is composed of components for designing the propulsion system performance, determining the system weight, generating a corresponding nacelle shape, and determining the shape of the aft plume structure. This assembly is a combination of both improved components and completely new capabilities. Additional components are available for plotting the generated data and for visualizing geometries. Components have also been developed to aid in automated linking to downstream components for ease of integration. The key components are described below.

4.2.1   Numerical Propulsion System Simulation (NPSS)

Numerical Propulsion System Simulation (NPSS) is the principal code that is exercised in propulsion performance modeling and operates as a framework itself to integrate engine components that are characterized by external codes or derived from existing libraries (ref. 3). Within the larger MDAO framework, NPSS functions as a component; as such, the development of an improved NPSS-component interface was identified as a critical need for the FAP. Previous attempts to create a file-wrapped component yielded an NPSS-component wrapper that required tedious manual rework for each new engine model. Two new methods for integrating NPSS with ModelCenter were developed: a component plug-in and an AnalysisServer JavaBean (see ref. 2 for a description of component wrapping methods). These two methods are similar, but each offers some specific advantages. The component plug-in runs on the user's local (Windows) machine and can use the NPSS Visual Based Syntax (VBS), which is the NPSS graphical user interface, to display the NPSS model. The AnalysisServer JavaBean can be run on a remote, possibly more powerful, host. Both provide easy access from ModelCenter to the variables and files of the NPSS model. Only the component plug-in is used in the Gen 0 framework.

No changes to the NPSS code were envisioned for this milestone. During the development and user testing, the component plug-in worked well for the design of a new concept. However, when tested for system-level optimization, some issues were discovered that required modification of the NPSS code. In particular, the external optimizer can specify design points that are not physically realistic. As a result, the NPSS solver was enhanced so that it could handle a wider range of states. In addition, a multisession capability was added to the NPSS implementation; this capability enabled multiple instances of NPSS to execute in the same process model on the same server.

The principal advantages of the plug-in component are its speed and versatility. The primary performance difference comes from the fact that a file-wrapped component requires a complete restart of the code every time a new design point is evaluated, while the component plug-in can stay resident in computer memory. Additional performance gains come from passing variable values through memory rather than creating input files from templates and parsing the output files, as required with a file or script-wrapped component. The total performance gain can be significant when NPSS is part of a system-level optimization because the actual calculation for each design point is quite small compared with the overhead of restarting NPSS. In addition to the improved computer performance, the plug-in is much easier and more intuitive to set up for a new concept model.

The new NPSS plug-in component is shown in figure 2. Multiple instances can be present in a single ModelCenter process model, and if needed, multiple ModelCenter sessions can be running at the same time.
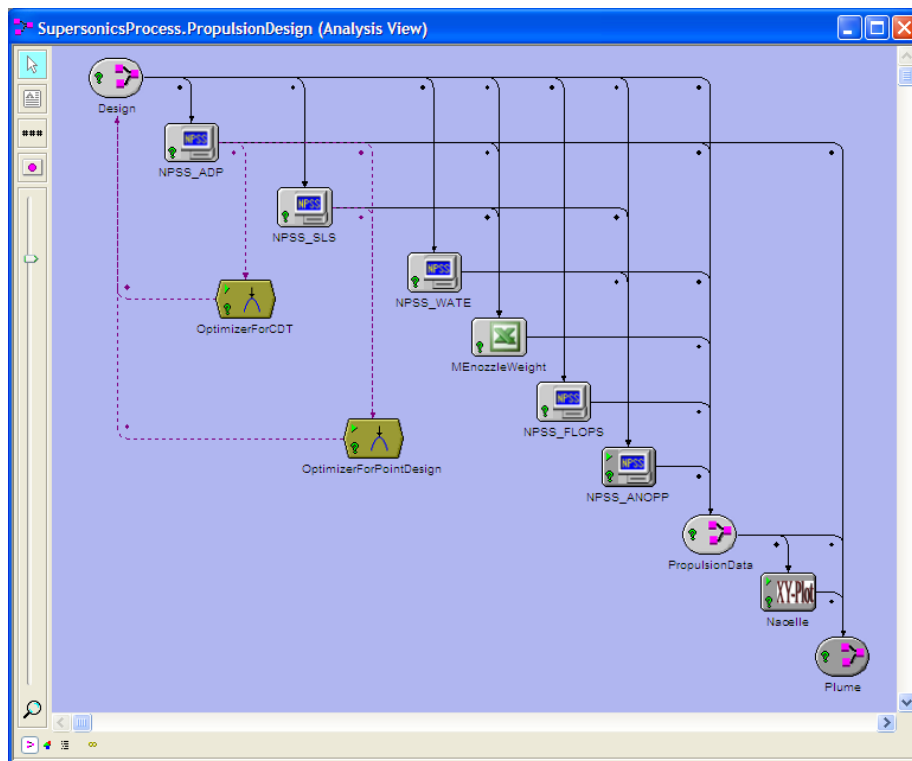

Figure 2. Series of NPSS components in the propulsion assembly.

One particularly useful feature of the NPSS plug-in is the graphical user interface (GUI), which allows the user to access any variable in the NPSS model and assign it to variables in the local ModelCenter workspace (see figure 3). Figure 3 also indicates how to access this plug-in GUI through the ModelCenter interface.
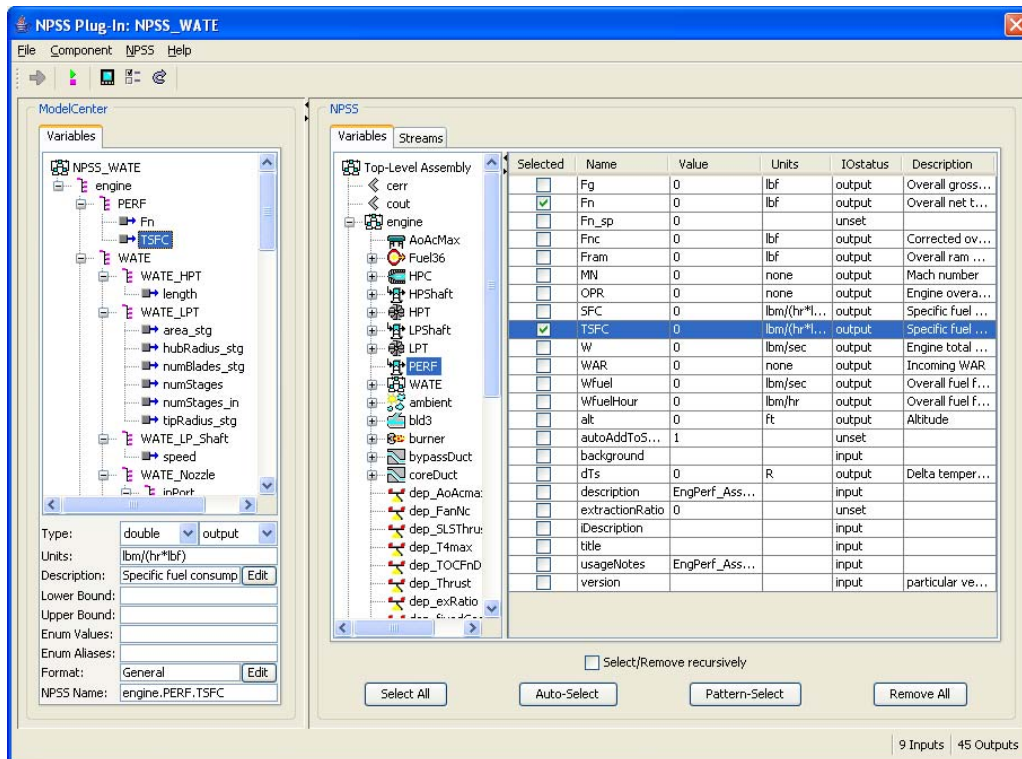
Figure 3. GUI for NPSS plug-in in ModelCenter.

Note that NPSS as a standalone code also has its own built-in GUI, called VBS. This is a completely separate entity from the plug-in GUI and should not be confused with it. The ModelCenter plug-in GUI and the NPSS VBS are partially integrated; VBS can be displayed from the plug-in GUI which allows the user to interact with the NPSS model via VBS. However, simultaneous updates via VBS and ModelCenter can interact in undesirable ways. For now, users are encouraged to avoid VBS for this reason but are encouraged to use the plug-in GUI instead. VBS is still quite useful for visualization of the NPSS model data-flow diagram, so the ability to access it from the plug-in is a desirable feature.

4.2.2 Weight Analysis of Turbine Engines (WATE++)

The size, shape, weight, and location of the turbine engines are critical data for the conceptual design of supersonic aircraft. The designer can input the propulsion location, but the other information must be estimated based on the design specifications for the engine. Weight Analysis of Turbine Engines (WATE) is a computer code that is employed to estimate the weight and dimensions of gas turbine engines; the code was originally developed by the Boeing Military Aircraft Company in 1979 and is described in reference 4. The computer code calculates the weight and dimensions of engine components using primarily semi-empirical methods that can be augmented with analytical calculations for specific component elements. For example, structural finite element method (FEM) analysis is available for turbomachinery disks. This code provides an accuracy of approximately ±10 percent of engine weight for existing state-of-the-art engines. For new engine architectures, the capability and accuracy of the WATE program requires more physics-based and rule-based component design methods to provide greater conceptual-level geometry details and subsequent insight into technology and materials advancements. One such improvement that was executed under the FAP was the recoding of WATE in the C++ language (renamed WATE++), which allowed direct integration with NPSS and access to thermodynamic variables and NPSS utilities, such as the solver for mechanical flow path design.

The geometry information that is available in NPSS is one-dimensional and comprises flow areas and empirically derived lengths for the various components and subcomponents of the engine. While the WATE++ geometry details are being improved, many component capabilities that are required for supersonic propulsion, such as inlet design and mixer ejector nozzle design, are still evolving. Currently, the WATE++ code provides only a rough outline of the nacelle outer mold line (OML), which is provided in part by the former McDonnell Douglas Corporation and is generally not adequate for high- or even low-fidelity analyses for supersonic concepts. As a result of these limitations, the current process, which is described in the following section, employs simple rules for generating the nacelle OML based on assumptions regarding the requirements for structural integrity and aerodynamic efficiency of the nacelle.

Given that the WATE++ code is already directly integrated with NPSS, the NPSS component plug-in is able to expose all the WATE++ variables for any NPSS model that includes a weights analysis. A key remaining challenge is to improve the interface between NPSS/WATE++ and other MDAO components, such as aerodynamics, flight dynamics, and community noise. The interface includes all the geometric and thermodynamic data that are needed for community noise predictions (section 4.9.3). Similarly, the engine fuel burn and combustor emissions characteristics are needed at an aircraft level for assessing environmental impacts throughout the mission (section 4.7). While the interface to community noise and aerodynamics is relatively complete in the Gen 0 framework, more work is needed to improve the robustness of the existing interface (section 4.2.3, for example) and to establish new links to other disciplinary codes. For example, the influence of deliberate and incidental propulsion thrust-vectoring impacts needs to be linked with vehicle maneuvers, even at a low fidelity, to accurately model critical commercial supersonic vehicle issues, such as takeoff angle of attack and wing aeroservoelastics during cruise. To enable higher fidelity analyses, such as CFD and structural FEM, watertight surface representations at a sufficient level of geometric detail will be required from NPSS/WATE++. Additionally, multidimensional thermodynamic flow properties are necessary to initialize CFD boundary conditions (methods for expanding and recompressing the current one-dimensional NPSS). These capabilities are presently under development in cooperation with discipline experts and will be incorporated into the existing MDAO framework in future incremental releases.

4.2.3   Nacelle Geometry

Because the NPSS/WATE++ codes currently have no supersonic inlet or mixer ejector nozzle design capabilities, a manual redesign of the nacelle OML was needed for each engine cycle, usually based on three or four control points, or diameters. This gap was partially filled during the development of the Gen 0 framework. In the current framework, the nacelle OML is based on turbomachinery geometry from WATE++ and inlet and nozzle length-to-diameter ratios. These ratio values for the inlet and nozzle are inputs to the process and are primarily based on previous designs. The inlet capture area and nozzle exit area are known from cycle analysis. The propulsion data assembly in ModelCenter builds the nacelle geometry (see figure 4) based on the following assumptions:
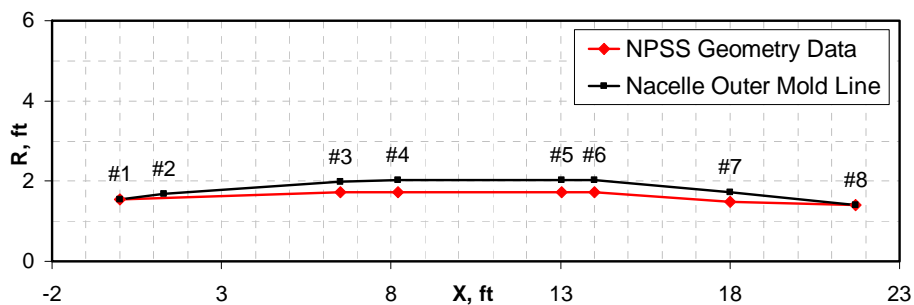

Figure 4. Low-fidelity nacelle OML.

- #1   At the inlet entrance, the radius is based on the capture area.

11

- #2   At 20 percent of the inlet, the radius is based on an initial lip angle of 6 deg.
- #3   At the fan face, the radius is based on the fan diameter plus containment plus 15 percent.
- #4 to #6   From the fan exit to the nozzle entrance, the radius is based on the fan diameter containment plus 17 percent.
- #7   At the nozzle midpoint, the radius is the larger of the bypass duct tip radius plus 17 percent and the average of the maximum (#6) and nozzle exit radii.
- #8   At the nozzle exit, the radius is based on the nozzle exit area.

This rudimentary design capability to generate a credible and aerodynamically efficient nacelle OML in a robust and automated manner removes the need for a manual redesign of the nacelle, which facilitates the parametric trade studies that are discussed in section 5. This method works well because the nacelle is relatively aerodynamically efficient, the mixed-flow turbofan (MFTF) propulsion system design cycle is fixed, and the perturbations to the cycle design parameters are relatively small. Higher bypass ratio engines or other more complex (variable) cycles will likely require additional design detail.

4.2.4   Plume Shape Prediction

The ability to produce a quick estimate of the plume shape is one of the most notable advances in the Supersonics Project. The plume shape calculations use NPSS calculations for the nozzle throat and exit areas and the corresponding flow properties. Additional nozzle geometry, such as internal and external flap lengths, are input. The low-fidelity nozzle plume profile or boundary is computed with the assumption of one-dimensional ideal gas flow (refs. 5 and 6). The implemented computer model uses a variable specific heat ratio γ that is based on temperature and accounts for the effect of the nozzle boattail on the local static pressure by assuming a Prandtl-Myer expansion. The model includes factors to adjust how the initial "average plume slope" is determined, the degree to which the nozzle 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 reference 5. The model begins by determining 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_∞, which can be affected by the nozzle boattail, then the nozzle is underexpanded. If $P_e < P_\infty$, then the nozzle is overexpanded. For underexpanded nozzles, an oblique shock is created by the plume; this shock affects the local free-stream pressure, thus creating the need for an iterative process.

For underexpanded flow, a modified version of the method proposed by Nash, Whitaker, and Freeman (ref. 5) is used to predict the plume shape. 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 reference 5, so the process is only summarized here with emphasis placed on the differences. The initial plume boundary angle α_1 is determined through iterations using the following equation:

$$\alpha_1 = \nu_1 - \nu_{noz} + \theta$$

where ν_{noz} is the Prandtl-Meyer angle that corresponds to the jet Mach number and ν_1 is the Prandtl-Meyer angle that corresponds to the nozzle pressure ratio (P_t / P_∞). 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 α_1 is varied until the pressure at P_2 reaches equilibrium. The method in reference 5 has been modified to account for a nozzle boattail angle β and the ratio of the specific heats γ. The local pressure in the vicinity of the nozzle exit is reduced by the Prandtl-Meyer expansion of the flow around the nozzle boattail. The boattail angle can have a significant impact on the plume shape, 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 affect 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 β and α_1 and only when $\beta > \alpha_1$.
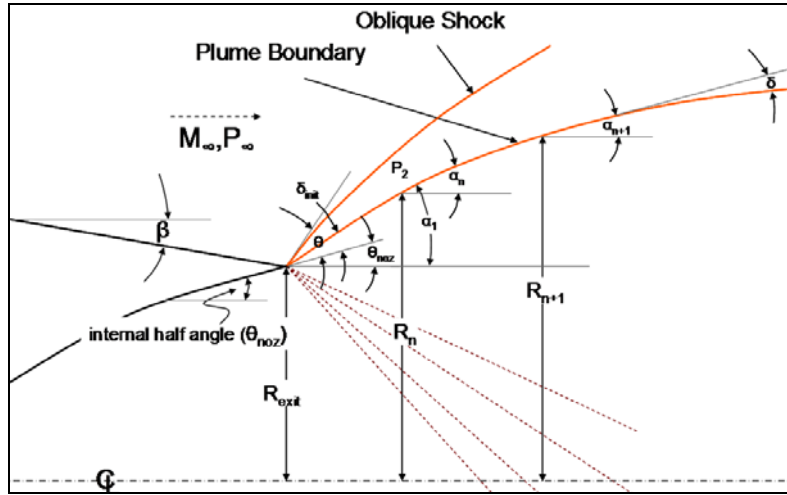

Figure 5. Geometry of plume for an underexpanded nozzle.

For overexpanded nozzles (see Figure 6), 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 $\Delta\nu_1$ is computed from the Prandtl-Meyer expansion of the flow from P_∞ to P_{avg}. The nozzle flow is turned by δ 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 conservation of mass. A new P_{avg} is then computed from the following equation:

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

13

where the static pressure ratio (P2/P1) is computed from the oblique shock relation by using the axial component of the Mach number and θ; 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 Δv until the Prandtl-Myer angle is equal to zero. The initial jet Mach number is the axial component and is updated at each step based on 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; these will be addressed in subsequent FAP activities in collaboration with discipline experts.
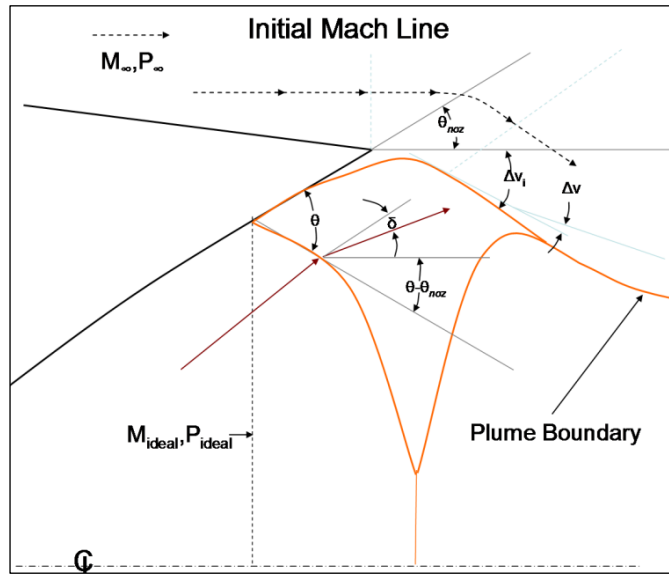
Figure 6. Geometry of plume for overexpanded nozzle.

4.3   Geometry

Modeling the geometry is a major challenge for a multifidelity MDAO system. A balance must be struck between conceptual-level parametric geometry and the highly detailed CAD geometries. Either of these geometry approaches can be effective for a given level of fidelity; however, for automating variable fidelity capabilities, the repeated conversion from component level geometries to smooth, watertight OMLs is quite difficult. The present MDAO framework contains integration and analysis tools that can be used to address this challenging problem. Efforts in this area are underway and should strengthen this capability in future releases. The current integrated capabilities are described below.

4.3.1   Vehicle Sketch Pad (VSP)

Vehicle Sketch Pad (VSP) is a high-level (i.e., low- to medium-fidelity) aircraft geometry layout tool that provides the parameterized or discrete pointwise geometry representations that are required by other components. Typically, VSP parameters are simple descriptors of the basic geometry (e.g., wing span) and can be used as design variables in a system-level optimization. The VSP tool provides wire-frame and solid modeling of the full aircraft and produces detailed geometries that can be used by various aerodynamics or structures codes. In addition, the tool allows a visual inspection of vehicle designs as they are being constructed (see Figure 7). The VSP tool was improved to provide the capability for exporting geometry formats that are required by the aerodynamic components. Additional import and export capabilities are planned for the future.
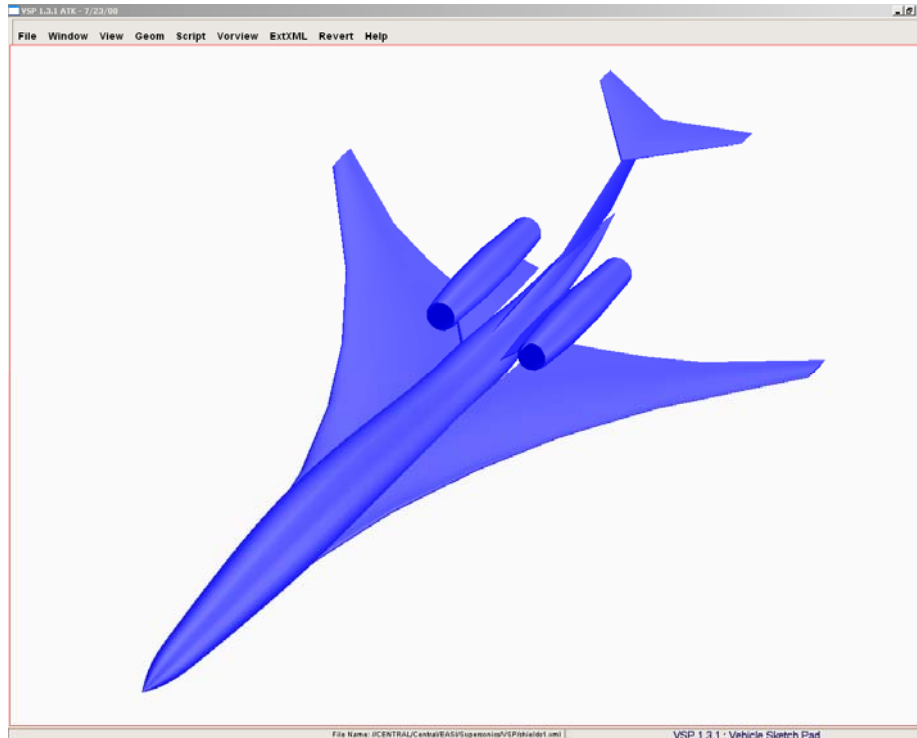
Figure 7. View of Supersonic Business Jet in VSP.

The VSP tool is included in ModelCenter process models through the VSP plug-in, which was originally developed under the CDS project (see Figure 8). The plug-in allows users to load the geometry of an aircraft concept. When an input value changes and a downstream component needs geometry data, the plug-in runs VSP in batch mode. The VSP plug-in generates Hermite or Craidon-formatted geometry files and updates the output variable values (e.g., areas, spans, and volumes). Modifications to the plug-in improved the interfaces between VSP and other components in the framework. While the capabilities of the VSP tool and plug-in have increased marginally, the capabilities of the supersonic process model have increased significantly.
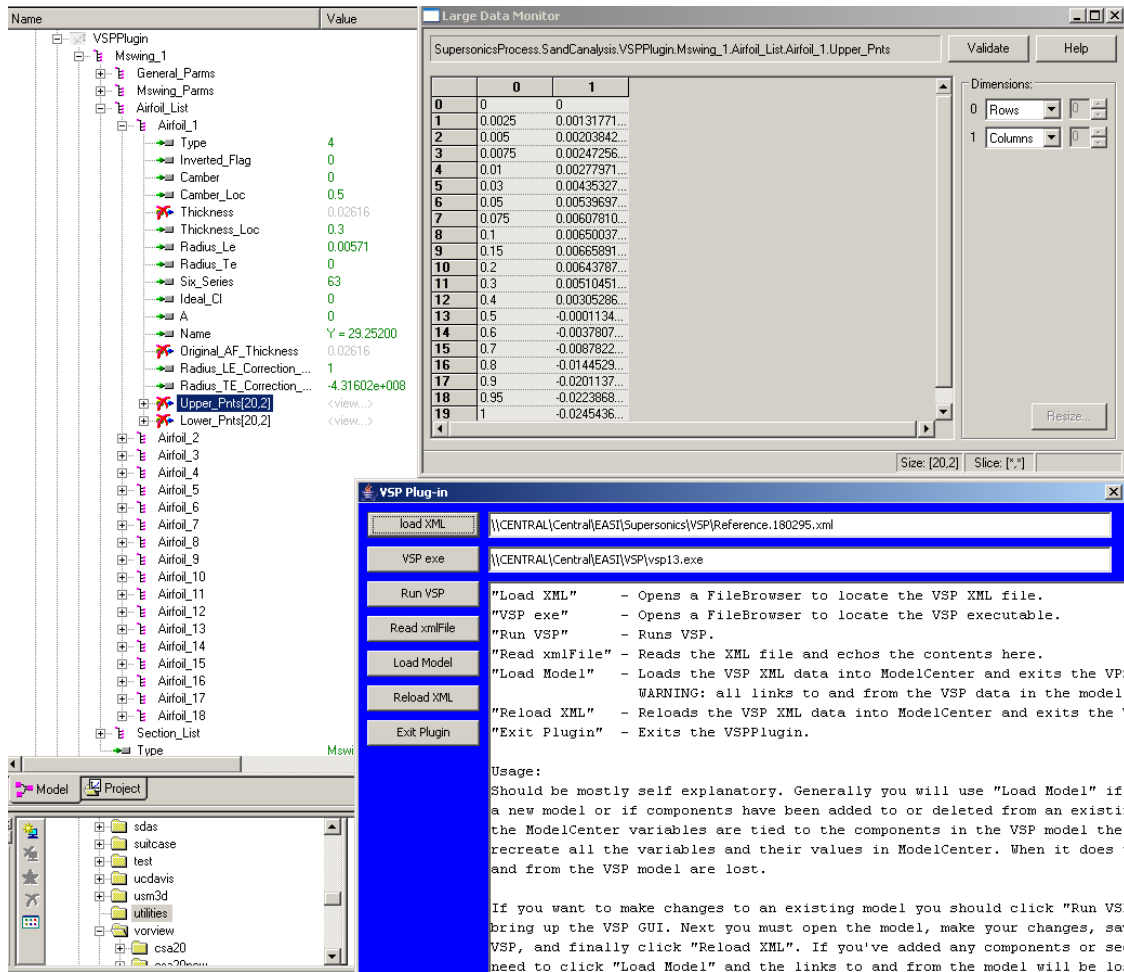
Figure 8. View of VSP plug-in.

4.3.2 CFD Grid Creation

A major headache for systems analysts is the need to convert conceptual geometry for use in high-fidelity analyses, such as FEM or CFD analysis. That capability gap was partially closed during the development of the Gen 0 framework. The framework can now convert conceptual-level geometry that is produced by VSP into a watertight surface geometry and then automate the generation of an initial CFD volume grid. This capability has been tested for several VSP models that include fuselage, wing, canard, pylon, nacelle, horizontal tail, and vertical tail components. This cutting-edge capability is illustrated in Figure 9 and Figure 10.

The key technical challenge is to produce a volume grid without human intervention. The approach is to import a conceptual level geometry and use a new tool called iPatch to patch any regions in which the aircraft components, such as fuselage and wings, do not intersect properly. Once a watertight geometry file is available, then the codes AutoSrc and VGRID (ref. 7) can produce an initial volume grid. The quality of this grid is determined by the locations of the lines and sources that influence the density of the mesh. The AutoSrc code places sources along the centerline of the fuselage, along constant percent-chord lines on the wing, and around the inlet and exit perimeters of the nacelle. All sources are based on characteristic lengths of each geometry element and will scale properly if the vehicle's shape changes. In the Gen 0 framework, a set of sources can be placed automatically, and a three-dimensional mesh will be produced. This is a significant improvement over past capabilities.

The aim of the CFD grid-creation subprocess is to automatically create grids that are good enough to ensure a converged CFD solution. The process must be fast and robust. In addition, grids for different Mach numbers and angles of attack must be able to be confidently produced once an acceptable baseline grid is available. The current tools have been tested on a number of supersonic concepts. This evaluation proved that nonexperts can produce preliminary CFD results. However, manual inspection of the mesh and intelligent modification to the source locations are necessary to produce high-quality aerodynamic performance calculations. Even greater care is necessary to produce grids for sonic boom predictions. These are areas for future research and testing.
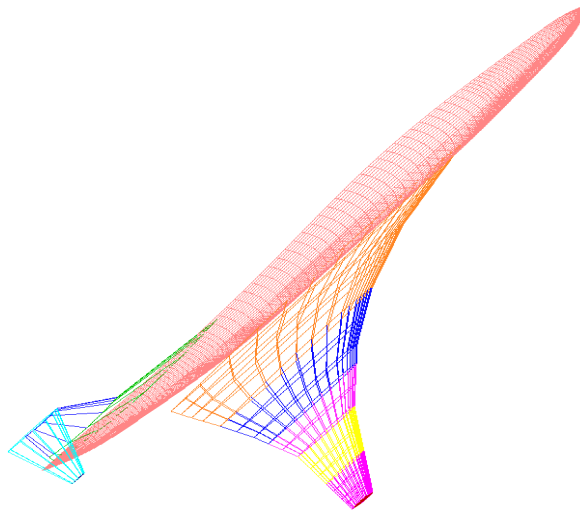
Figure 9. Conceptual geometry with gap between wing and fuselage.
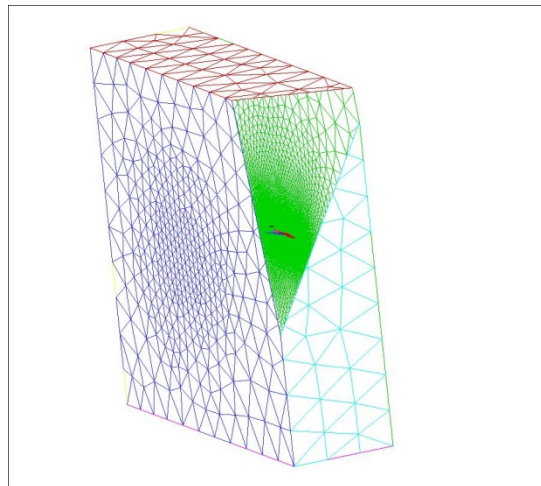

Figure 10. Typical volume grid.

4.4 Aerodynamics: Design and Analysis

The aerodynamic design subprocess is shown in Figure 11, and the aerodynamic analysis sub-process is shown in figure 12. In addition to the design and analysis components, both assemblies include a number of pre- and

postprocessing and automated plotting components. A brief description of the design and analysis components that are used in these assemblies follows.
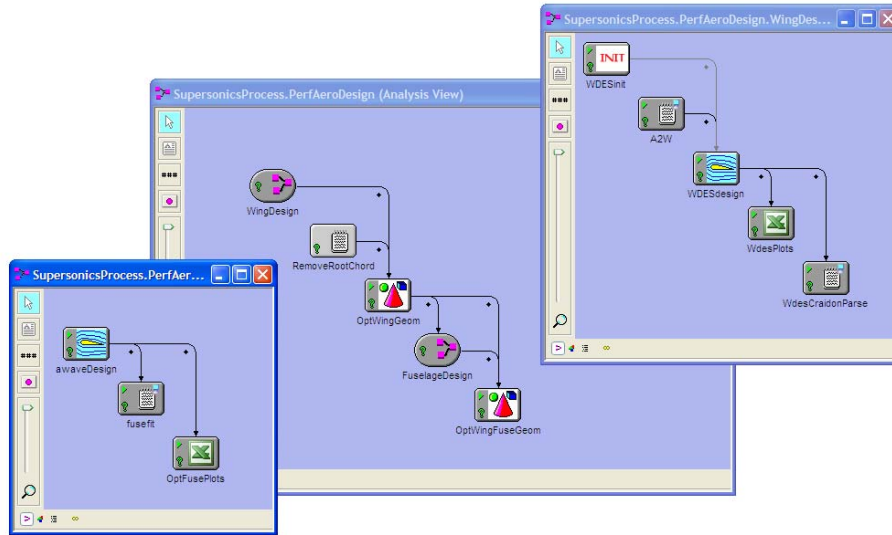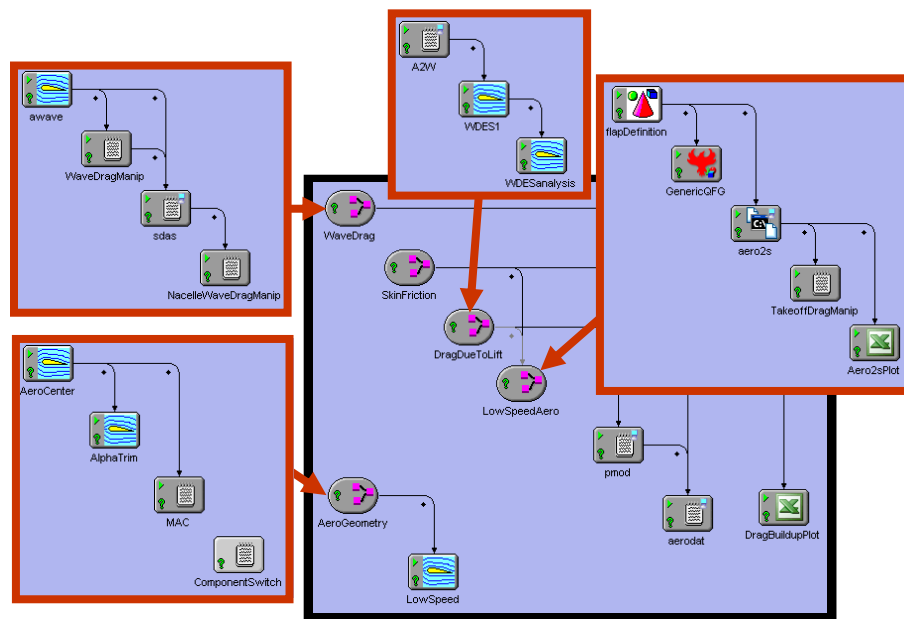

Figure 11. Aerodynamic design subprocesses.

Figure 12. Aerodynamic analysis subprocesses.

4.4.1   Supersonic Cruise Efficiency

The low-fidelity high-speed aerodynamics analysis and design capability was created and tested during the CDS project. It required very little modification to meet the needs of the Gen 0 framework. The existing capability consists of components for wave-drag, skin-friction/form/roughness drag, and drag due to lift. For the wave-drag analysis, the AWAVE code is used. AWAVE is a streamlined, modified version of the Harris far-field wave-drag program (ref. 8). The AWAVE code has all of the capability and accuracy of the original program plus the ability to

18

include the approximate effects of angle of attack. For drag due to lift, the WINGDES code (ref. 9) is used in both design and analysis mode. In the design assembly, an optimum wing camber can be designed for a given lift coefficient and other design parameters. In the analysis assembly, WINGDES is used to generate polar data throughout the entire speed regime. Entire mission flight polars can be generated very quickly to be passed along to the mission performance code. For a multifidelity implementation, a few cruise points are evaluated with a high-order code, and then the lower fidelity results are calibrated accordingly. The best way to implement this approach is currently being studied under an NRA contract and has not been implemented in the current system.

The original CDS supersonic capability was adequate for assessing a baseline configuration but a few improvements have been made. For example, originally the number of fuselage cross-sections had a fixed size and were limited to a relatively small number (30) of cross sections. These limitations have been removed in the Gen 0 framework. Now the number of cross sections is an input value that can be varied during the design process. Moreover, the capability has been tested with a large number of fuselage cross sections and more detailed wing and tail designs. These improvements are necessary so that the aerodynamic subprocess can interface well with the sonic boom analysis.

4.4.2   Low-Speed Aerodynamics

A low-fidelity low-speed aerodynamics capability was also created and tested during the CDS project. This capability uses the wing analysis computer program AERO2S that is described in reference 10. Attainable leading-edge thrust, which is calculated as described in reference 10 is taken into account. The code, developed by Harry W. Carlson, provides a subsonic aerodynamic analysis for a wing with flaps in combination with a second lifting surface (a canard or a horizontal tail). A three-surface version is also available and could be added to the system with little additional effort. The AERO2S code is applicable to the analysis of lifting surfaces under conditions that tend to induce separation and degrade performance, as well as promote attached flow. Low-speed polars that are generated from this method are currently being used for takeoff performance analysis. An interface has been developed to help rapidly lay out flaps and visually inspect the defined locations, as well define the flaps so that they are automatically adjusted for planform variations. Many more process improvements could be added to the existing framework in this area, to include wind-tunnel data correlation, trim analysis, and ground effects (currently, the ground effects are accounted for using a method that is integrated in the takeoff performance analysis tool). Many of these manual processes exist but have not yet been integrated. These methods could be added to the model to improve the fidelity of the low-speed analysis process if time and resources are available to develop the tools and methods.

4.4.3   CFD Analysis

The high-fidelity CFD analysis assembly currently consists of the USM3D CFD code (ref. 11). This capability provides not only higher fidelity aerodynamic performance data but also better lift distributions and pressure distributions for use in sonic boom analysis. The current SUP MDAO framework can convert conceptual geometry to CFD geometry and then generate a CFD grid automatically (as discussed subsection 4.3.2) for use with USM3D. The USM3D code has been wrapped for use in ModelCenter, but the process automation has not been completely debugged as a result of the long computer times that are required for each USM3D analysis. The integrated automatic capability still requires further testing to ensure robust operation without user intervention and should be available in the next release of the system. Additional CFD codes are also being evaluated for integration and testing in the near future.

4.5 Low-Boom Design

The low-boom design capability is based on far-field theory for boom propagation, which assumes that the full configuration can be treated as an equivalent body of revolution in the sonic-boom analysis. Three key components make up the low-boom design process: (i) a computer code HYBRID (ref. 12) to generate a target equivalent area (A_e) distribution for a low-boom ground signature based on the George-Seebass-Darden boom minimization theory (refs. 13 and 14); (ii) a computer code (PBOOM) to calculate the total A_e distribution of a given configuration (ref. 15); and (iii) a design optimization code (BOSS) to reshape the fuselage for boom reduction (ref. 16).

The current low-boom conceptual design process determines a layout for all of the components (fuselage, wing, tails, pylon, nacelle, and canard), designs an aerodynamically optimal wing that satisfies the mission requirements, and then reshapes the fuselage (by modifying the discrete radius distribution, as shown in Figure 13) to achieve a low-boom configuration. Figure 14 shows the ModelCenter layout of the analysis and the optimization tools that are used in the low-boom conceptual design process.
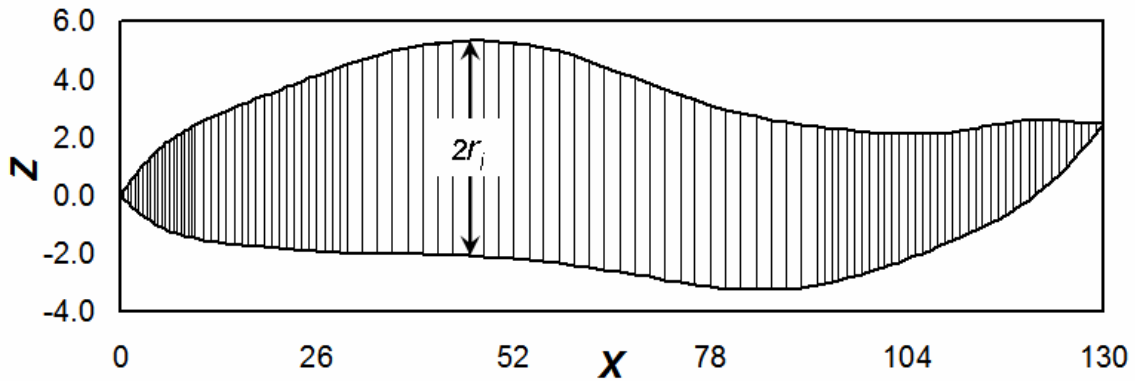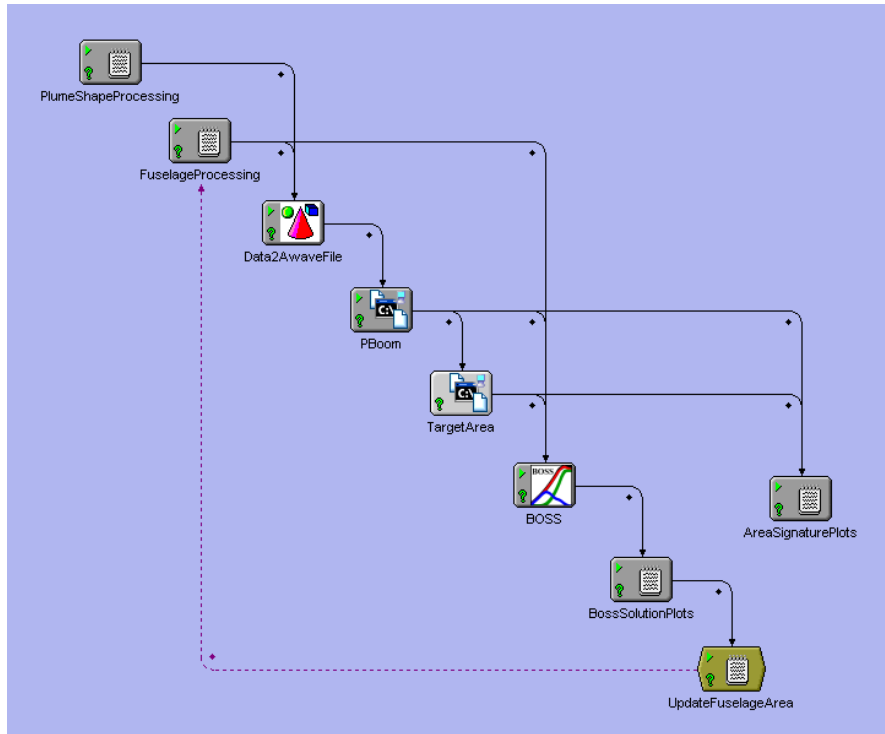

Figure 13. BOSS shaped fuselage.

Figure 14. Subprocess showing a boom-shaping iteration.

For any given supersonic concept (with wing, fuselage, nacelles, tails, and canards), a designer can examine the differences between the design and the target equivalent areas, decide which part of the design equivalent area curve needs to be modified, choose a desirable rate for the reduction of the discrepancies over the specified range, and select a parameter for smoothness control of the fuselage shape. In a matter of seconds, the BOSS component generates a fuselage shape that is based on the designer's inputs. If the generated shape is not acceptable, the designer can work on a different part of the equivalent area curve, change the rate of reduction, or relax the smoothness control until a desirable solution is found. The new configuration is analyzed by PBOOM to determine whether it has an acceptable low-boom ground signature. If not, the designer can use BOSS to further reduce the differences between the design and the target equivalent areas until the configuration has an acceptable low-boom ground signature. Using BOSS and PBOOM, the designer can generate a realistic, smooth fuselage shape that results in a supersonic configuration with a low-boom ground signature in a few hours. In addition, a designer can use BOSS to quickly eliminate any configuration that cannot achieve low-boom characteristics with fuselage shaping alone.

Note that if the previously discussed NPSS nacelle definition and plume shape are included in the A_e calculation, then the generated low-boom concept will account for plume affects in the ground signature. Moreover, for low-boom design at a higher fidelity level, the A_e calculation will be based on CFD geometry and a CFD-based lift distribution.

In the following three subsections, brief descriptions of the three key components in the low-boom design process are provided.

21

4.5.1 HYBRID

The HYBRID code is used to generate an optimum target area distribution for minimum sonic boom. For a given beginning cruise weight, altitude, Mach number, design initial overpressure, and two parameters that define the ground signature shape (the length of the flattop and the slope of the ramp), the HYBRID code can generate a target A_e distribution for a configuration with a conic fuselage nose and minimum effective length. When the other parameters are fixed, as the length of the flattop is increased, either the effective length increases or the initial overpressure increases. See figure 15 for examples of the flattop, ramp, and other low-boom target ground signatures that are typical of those commonly used for the low-boom design. The current low-boom design process is to find a supersonic concept for which the A_e distribution matches the target A_e distribution (generated by HYBRID) as closely as possible. More details of HYBRID can be found in reference 12. Note that these low-boom target ground signatures might not be achievable by practical configurations. More flexible methods are needed for generating low-boom target signatures that accommodate aft shaping of the ground signature. New capabilities for defining these alternate target distributions are currently being developed within the SUP project and will be available in the next release of the MDAO framework.
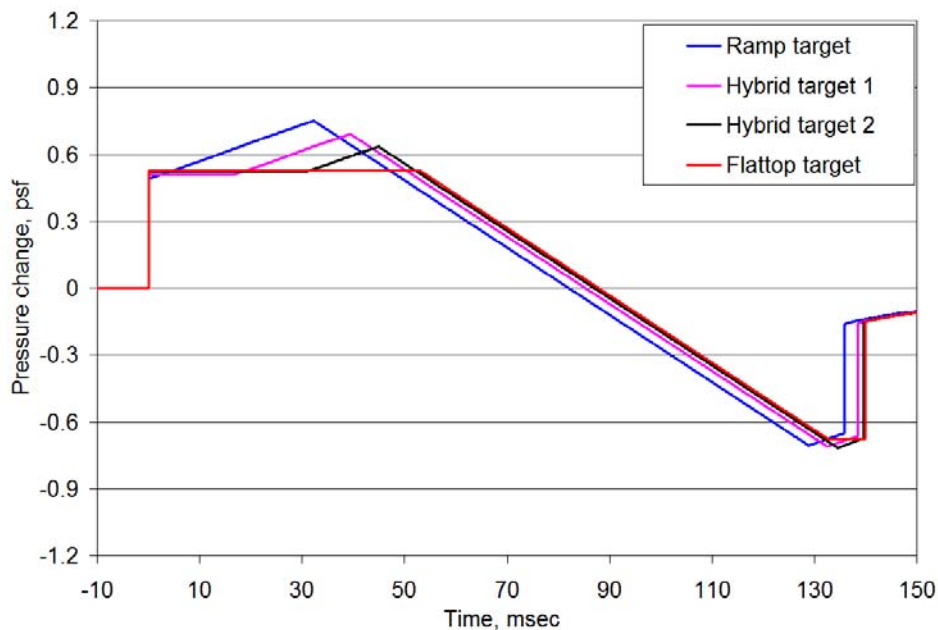

Figure 15. Typical types of target sonic boom signatures.

4.5.2 PBOOM

PBOOM is an integrated computer program that is used to predict the sonic boom characteristics of supersonic vehicles throughout the flight profile. PBOOM uses a single geometry description and combines the equivalent area and Whitham F-function methods with a signature propagation method. The aircraft geometry is input in the Craidon geometry format from VSP. The equivalent area that results from volume, lift, and interference is calculated by PBOOM with a modified linear aerodynamic method. The F-function is calculated based on the total equivalent area distribution, and the equivalent sonic boom signature is then propagated to the ground. PBOOM provides a quick approximation of the sonic boom ground signature based on the input geometry and is the current tool of choice for this prediction at the low-fidelity level. See reference 15 for more details of PBOOM. Efforts are

currently underway to deconstruct this integrated code and implement each analysis component with individual codes. The capability will then be reintegrated with the ModelCenter framework, thus enabling the individual methods that are currently in PBOOM to be replaced with upgraded tools.

4.5.3   *Boom Optimization using Smoothest Shape Modification (BOSS)*

The BOSS component has recently been developed, and details can be found in reference 16. The fuselage shaping for low-boom design with BOSS is an interactive optimization process that involves trade-offs between maintaining a realistic smooth fuselage and reducing the discrepancies between the design and target A_e distributions, while trying to maintain acceptable aerodynamic performance.

Two key user control parameters for BOSS are the smoothness parameter s (0 for no smoothness requirement and 10 for using the smoothest shape) and the desirable rate of reduction ρ of target-area matching error over a specified effective distance range. If ρ and s are smaller, then the matching and required smoothness are easier to achieve. A maximum execution time of about 3 min is also implemented by setting a default value of 200 for the maximum number of iterations for each BOSS run. As a result, BOSS terminates quickly with either a desirable solution or the best solution possible. This implementation allows a user to experiment with various choices of effective distance range (X_{start} and X_{end}), ρ, and s to close the gap interactively between the equivalent area distribution of the configuration and the target. If a BOSS run fails to find a desirable solution, then either no further improvement can be made in the specified range or any further reduction of the matching error may require the use of a fuselage shape that is less smooth.

For any given wing planform and layout of aircraft components, BOSS reduces the design time of low-boom supersonic concepts from months to hours. More importantly, BOSS allows a quick closure of the fuselage shaping process because BOSS lets the designer see the degree of deterioration of the fuselage shape that is necessary to further reduce the discrepancies between the design and the target A_e distributions. Figure 16 compares the target and actual A_e distributions, with a breakdown of the total distribution into distributions for lift, volume, and fuselage; figure 16 also shows various low-boom signatures and the ground signature of an actual low-boom design.
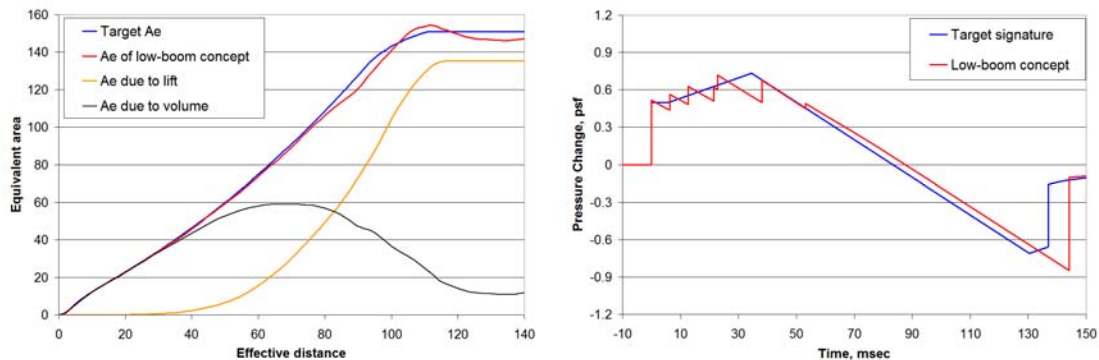

Figure 16. Area and signature plots.

4.6　Weights Analysis

Aircraft weights analysis is challenging for supersonic aircraft. For lower fidelity calculations, empirical aircraft weights estimates are traditionally used. Low-fidelity estimates tend to be inaccurate because of the lack of an existing weights database for these unconventional vehicles (primarily commercial supersonic aircraft). With higher fidelity structural analysis methods, better estimates of sized structural weights for a well-defined structural layout are possible, however, high-fidelity estimates are inaccurate because these do not include the as-built weights (e.g., doors, windows, secondary structure, and fasteners). A multiplication factor can be used on the high fidelity sized structural weights for system-level trades, but the absolute levels are still not well established. The Gen 0 SUP MDAO system was developed using the currently avavilable empirical and semi-empirical weights methods. Significant future work in this area is planned for the next few years; the results from this work are expected to be included in the next release of the system. The exact approach to overcoming this challenge is still being researched but will attempt to bring the higher fidelity analysis capabilities into the system level design loop by developing a hybrid structural and as built weights approach. The following sections describe the methods that are included in the current system.

4.6.1　Empirical Weight Estimates

The aircraft sizing and synthesis code FLOPS includes an aircraft weight estimation capability (ref. 17). The primary weight estimation routine utilizes statistical weight-estimating relationships that are derived from an optimization-based curve fitting approach, based on data from traditional configurations. In addition, an analytically based wing weight estimation capability is used for supersonic configurations. The input for the detailed wing weight analysis includes detailed wing planform geometry, thickness-to-chord ratios at each wing section, and an estimated load path sweep at each wing section. A theoretically derived wing bending factor is calculated by numerical integration along the specified load path to determine the amount of bending material that is required to support the selected input load distribution. The user may select from a list of standard load distributions (i.e., elliptical, triangular, and constant) or may input a user-defined load distribution. The wing is treated as an idealized beam with dimensions that are proportional to the wing local chord and thickness. The bending factor is modified for aeroelastic penalties (i.e., flutter, divergence, and aeroelastic loads) depending on wing sweep, aspect ratio, degree of aeroelastic tailoring, and strut bracing, if any. These modifications are based on a curve fit of the results of a study performed using the Aeroelastic Tailoring and Structural Optimization (ATSO) code to structurally optimize a large matrix of wings (ref. 18).

The FLOPS component has been wrapped so that a weights analysis can be run separately from the FLOPS mission analysis to allow future integration of higher fidelity weight analysis codes. The weights component input comes primarily from the final geometry and the propulsion design assemblies. The output includes all subsystem weights, as well as centers of gravity and moments of inertia, for a set of user-specified loadings for use in stability and control analysis.

4.6.2　Point Design of Cylindrical Bodied Aircraft (PDCyl)

Point Design of Cylindrical Bodied Aircraft (PDCyl) is a semi-empirical aircraft weight estimation code that is intended for the conceptual design of aircraft (ref. 19). This code is appropriate for aircraft configurations that fall outside the historical database used by FLOPS, yet it avoids the high-resolution mesh and complex inputs required by structural FEM codes. Structural members are sized using fundamental structural principals, and the results are integrated to determine an estimate of the aircraft weight. As such, PDCyl is essentially a physics-based code that assumes an idealized structure to obtain an initial calculation for the weight and then performs regression to provide an estimation of the actual weight. Further details on PDCyl can be found in Ref. 19.

For the Gen 0 framework, a simple file wrapper was created for PDCyl, and a weights subprocess was developed, as shown in Figure 17. The PDCyl component contains 94 input variables, but default values can be used for all those except the 25 variables that are required to perform basic weight estimation. Default values (e.g. material properties and buckling coefficients) are based on a Boeing 747 model that is given in reference 19.
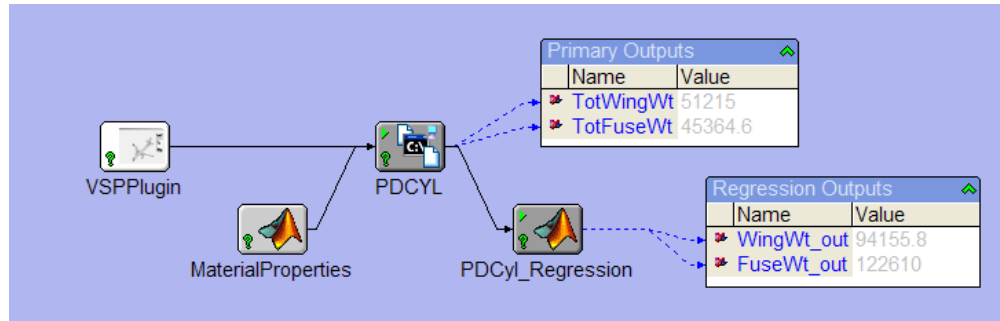

Figure 17. The PDCyl component in a ModelCenter process diagram.

The PDCyl component is usually linked to the geometry component and a MaterialProperties component, as shown in Figure 17. The MaterialProperties component generates all of the materials inputs, based on a material type that is selected by the user. This is one particular advantage of PDCyl over the FLOPS weights calculations in which the control of the materials is only available through the use of a few correlation variables, based on the percentages of the composite materials that are used and the amount of aeroelastic tailoring that is assumed.

Two issues surfaced while the component was being tested. First, the center-of-gravity (CG) calculation fails when the engine is located on the centerline of the fuselage. Although most current configurations do not have centerline engines, the code should be able to calculate weights for this case. This failure could indicate a larger, undiagnosed issue and may bring all results into question. Additionally, this limitation is not acceptable for supersonic concepts for which a centerline engine may be needed. A viable alternative for calculating the CG location is to use the CG build-up capability in FLOPS, and this will be done until the issue with PDCyl is resolved. In addition, wing and fuselage weights from various past supersonic concept studies were used to assess the applicability of PDCyl to structural arrangements that are typical of supersonic vehicles. This comparison raised additional doubts about the component. The lack of actual data for validation purposes poses a significant problem. As a result, the PDCyl subprocess has been included in the current process model but remains an item for further testing and validation based on the noted issues.

4.7   Mission and Takeoff Performance Analysis (FLOPS)

The aircraft sizing and synthesis code FLOPS includes mission analysis and detailed takeoff and landing capabilities (refs. 20 and 21). For the analysis of supersonic concepts, FLOPS requires propulsion system performance and weight data as described in section 4.2, geometry as described in section 4.3, aerodynamic data as described in section 4.4 and weights data as described in section 4.6. Aerodynamics analysis components along with the FLOPS component were created and tested during the CDS project. For the Gen 0 framework, the FLOPS implementation was enhanced to allow increased flexibility and processing speed. The new component includes methods that automatically remove input options that may not be needed for a given study or at all, in the case of supersonics, which reduces overhead (execution time) and declutters the overall model (see Figure 18).

The mission performance data is based on weights data from the weights analysis assembly, aerodynamic data from the aerodynamic analysis assembly, and weight and performance data from the propulsion system design assembly.

Based on energy considerations, optimum climb profiles may be flown to the start of cruise conditions. The cruise segments may be flown at the optimum altitude and/or Mach number for maximum range or endurance or to minimize emissions, at the long-range cruise Mach number, or at a constant lift coefficient. Descent may be flown at the optimum lift-drag ratio. In addition, acceleration, turn, refueling, payload release, and hold segments may be specified in any reasonable order. Reserve calculations can include flight to an alternate airport and a specified hold segment. For supersonic aircraft, sonic boom overpressures are computed along the aircraft track. FLOPS also calculates performance constraints such as time to climb.

FLOPS uses the same weight and propulsion system performance data and low-speed polars from the aerodynamics assembly to perform detailed, three-degree-of-freedom takeoff and landing simulations. FLOPS computes the all-engine takeoff field length, the balanced field length to consider one-engine-out takeoff and aborted takeoff, and the landing field length. The approach speed is also calculated, and the second segment climb gradient and the missed-approach climb gradient criteria are evaluated. Insofar as possible with the available data, all FAR Part 25 or MIL-STD-1793 requirements are met. The module also has the capability to generate a detailed takeoff and climb profile for use in community noise analyses.

The output data of the FLOPS mission and weights analysis components represent the fundamental results of the overall model, with the output providing detailed information on the characteristics and performance of the analyzed aircraft. These data are required for community noise, cost, and stability and control analyses, and they also supply mission segment data for sonic boom analyses.
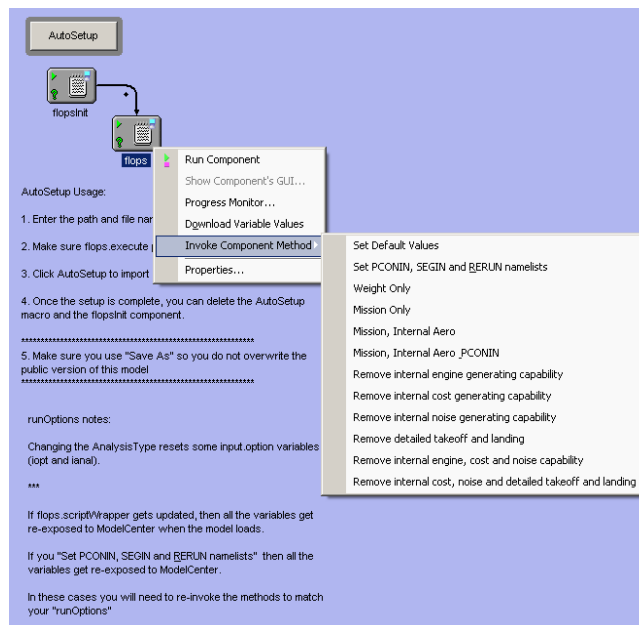

Figure 18. FLOPS component with new methods.

The Mission Performance module also has components that enable plotting of the mission flight data, engine decks, and other output tables that are generated by FLOPS. This capability within the system is comparable to the capabilities that exist in the GUI-based version of the standalone FLOPS code (XFLOPS).

4.8 Stability and Control

The Matlab™ Stability and Control Toolbox (MaSCoT) is used for conceptual-level stability and control analysis. MaSCoT was developed during the CDS project for performing low-fidelity assessment of aircraft stability and control for a set of flight critical conditions (FCC) shown in Table 3. When the CDS project ended, a stability and control tool with a set of basic capabilities was released. That version of MaSCoT performs a trim solution and a static stability analysis that calculates the equivalent stiffness and damping in the roll, pitch, and yaw directions and also calculates static margin, which is the distance between the aerodynamic neutral point and the aircraft center of gravity. Further detail on the original MaSCoT tool can be found in reference 22. The capability to perform dynamic stability analysis was added under the SFW project and is documented in reference 23.

Table 3. MaSCoT Flight Critical Conditions

No.	Code	Description
1	HN	Horizontal-Clean flight, Nominal status
2	HW1	Horizontal-Clean flight, + Wind Type1 status
3	HW2	Horizontal-Clean flight, + Wind Type2 status
4	HE	Horizontal-Clean flight, Engine Out status
5	HEW1	Horizontal-Clean flight, Engine(s)Out & + Wind Type1 status
6	HEW2	Horizontal-Clean flight, Engine(s)Out & + Wind Type2 status
7	TN	TakeOffRot-MaxLift1 flight, Nominal status
8	TE1	TakeOffRot-MaxLift1 flight, Engine(s)Out-Type1 status
9	TE2	TakeOffRot-MaxLift1 flight, Engine(s)Out-Type2 status
10	LN	LandingRot-MaxLift2 flight, Nominal status
11	LW1	LandingRot-MaxLift2 flight, + WindType1 status
12	LW2	LandingRot-MaxLift2 flight, + WindType2 status
13	LE	LandingRot-MaxLift2 flight, Engine(s)Out status
14	LEW1	LandingRot-MaxLift2 flight, Engine(s)Out & + WindType1 status
15	LEW2	LandingRot-MaxLift2 flight, Engine(s)Out & + WindType2 status

Figure 19 shows the ModelCenter process diagram, including the SandCAnalysis assembly and its contents. Three other components are required to generate the input to the MaSCoT component. The Vorview component generates the aerodynamic stability and control derivative input in the format that is required by the physics-based aerodynamic model. The RunCases component is a *driver* component that iterates with Vorview to generate the longitudinal and lateral stability derivative values for each of the three flight regimes (i.e., take-off, cruise, and landing) which can be analyzed by MaSCoT. The SandC component then assembles all the data into the input file that is required by MaSCoT. The 15 FCCs that are defined by MaSCoT and up to 5 user-defined FCCs are allowed, each of which can be individually activated in the SandC component.

The design of the SandCanalysis assembly overcomes several barriers. First, it gives the user control over which flight regimes to analyze. This control is useful if, for example, the user is only interested in takeoff FCCs or in simply sizing the horizontal tail. Second, it manages the flow of information between several different assemblies. For example, Vorview uses different geometry files for longitudinal and lateral stability derivative calculations. Thus, the VSP component in the FinalGeometry assembly needs to create the correct geometry file for Vorview. This flow of information is indicated by the solid line that feeds back from the SandCanalysis assembly to the FinalGeometry assembly. Additional forward links flow from WeightAnalysis for weight, geometry, and inertial data; from MissionAnalysis for Mach numbers and propulsion-system performance data for various flight regimes; and from FinalGeometry for geometric data, such as engine location.
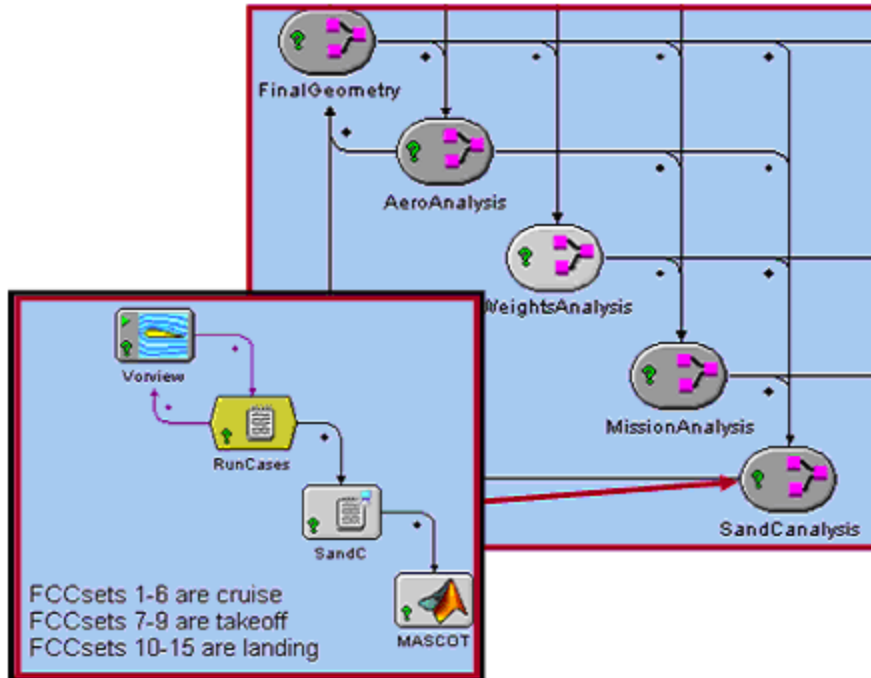
Figure 19. Stability and control process.

The output from the stability and control assembly is either a graphical or a numerical assessment of the stability characteristics at each FCC. The assessment includes a determination of whether the control authority that is available for trim is sufficient and whether the static stability characteristics are adequate. Control surface deflections and thrust levels are provided both numerically and graphically. See Figure 20 for a sample of output from MaSCoT for the supersonic business jet that was discussed in section 5.1.

MaSCoT still needs validation and added capability for supersonic concepts. For example, many supersonic concepts move fuel as part of their control authority, and MaSCoT does not currently have this capability.
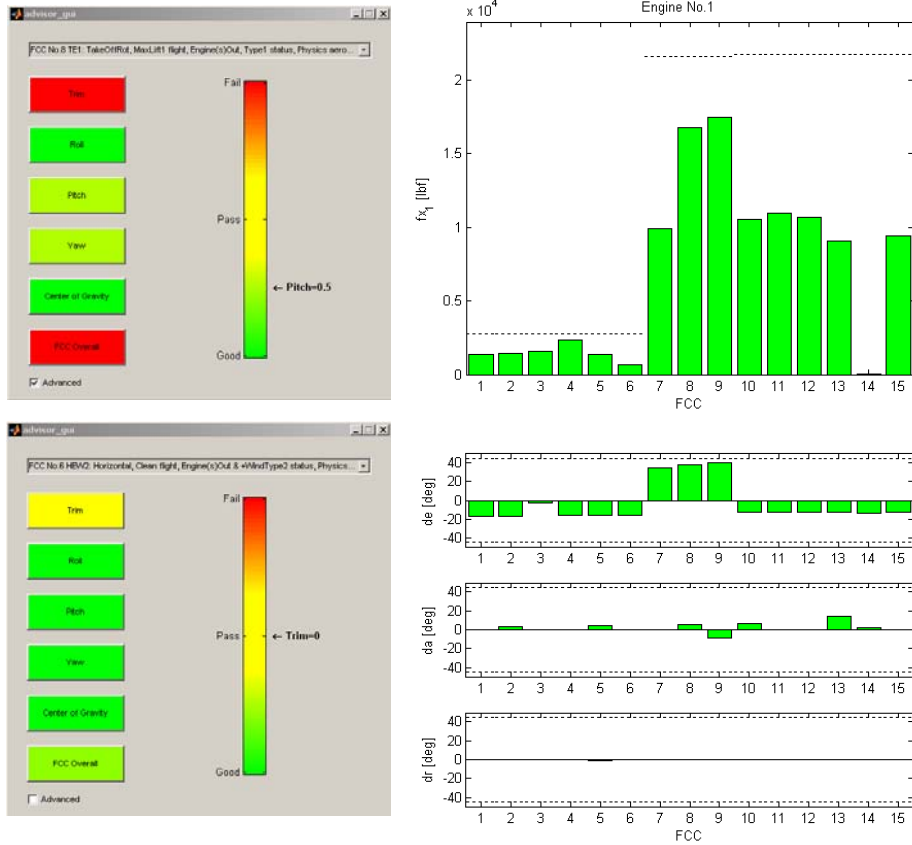
Figure 20. MaSCoT graphical interface.

4.9   Human Acceptance

Human acceptance is often subjective and can cover a wide range of disciplines from community noise impact, sonic boom, cost, and passenger comfort. Some aspects of human acceptance should become available through the stability and control analysis, such as lateral and longitudinal loadings and floor angles. Some other aspects of human acceptance, such as community noise and cost, are discussed in the following subsections. One aspect that is missing from the MDAO framework is cabin noise.

4.9.1   Sonic Boom Analysis

The integrated multifidelity sonic boom analysis process is shown in Figure 21. This analysis assembly allows sonic boom ground signatures to be calculated using either PBOOM in the low-fidelity mode or PCBOOM (ref. 24) in the higher fidelity mode. The path that is used depends on which prior analysis components have been exercised. The process includes the ability to automatically plot the resultant signatures either independently or together for comparison (if both low and high fidelity data have been computed).
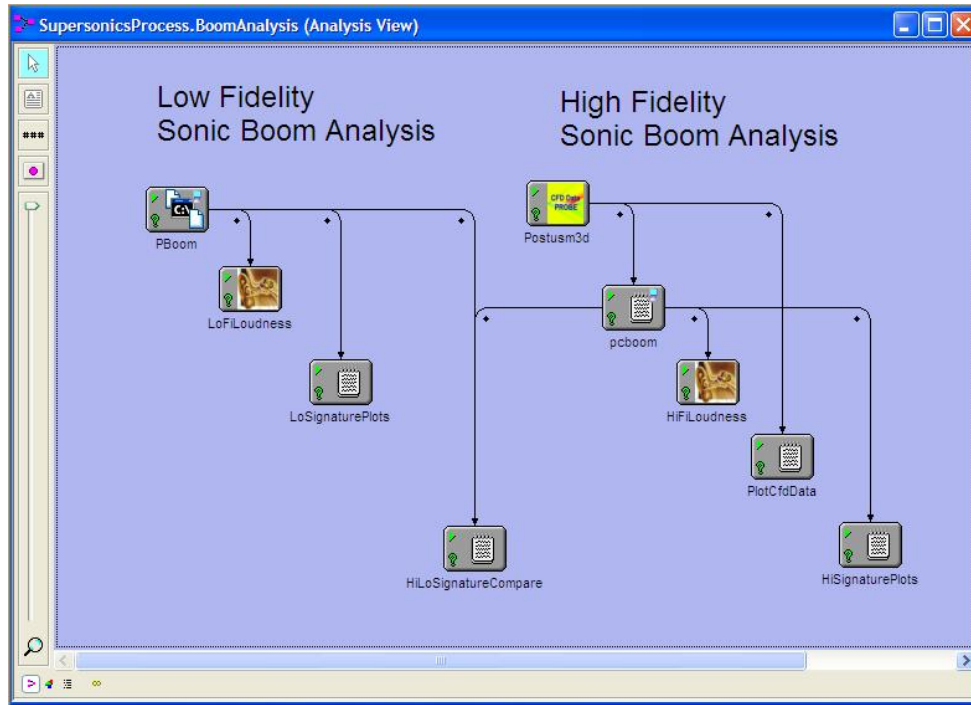
29

Figure 21. Sonic boom analysis process.

PBOOM was previously discussed in section 4.5.2. The code is used in the same way in both the design and analysis modules for computing the estimated sonic boom signature. In the boom analysis assembly, the actual data from the mission performance module for altitude and weight are used in the analysis.

PCBOOM is a sonic boom prediction method based on nonlinear propagation of a near or mid-field aircraft pressure signature. It can account for a horizontally stratified atmosphere and for ray focusing due to maneuvers. The input to the code is a normalized pressure distribution on a line segment that is parallel to and below the flight path. The sonic boom prediction proved to be sensitive to the radius at which the pressure inputs are calculated and to the quality of the grid and the grid spacing used for the CFD calculations of the near-field pressure. Some of the sensitivity may be corrected by using newer versions of the PCBoom code (e.g., PCBoom 4.70d is recommended rather than the PCBoom 4.1). Reference 24 describes several variants of the PCBOOM code and its capabilities.

4.9.2   Boom Loudness

Subjective loudness tests by Shepherd and Sullivan (ref. 25) and others have established the impact of sonic booms on people. These tests confirm a relationship between sonic boom signature shape and perceived loudness. Specifically, long front shock rise times and low maximum overpressure tend to be less annoying than other boom shapes. Shepherd recommends the Stevens Mark VII method (ref. 26) to predict the annoyance caused by a sonic boom. A ModelCenter component based on Stevens method and the subjective loudness tests of Shepherd and Sullivan is available in the MDAO framework.

4.9.3   Community Noise Analysis

The community noise assembly is centered around the Aircraft Noise Prediction Program (ANOPP). ANOPP was designed to predict the total aircraft noise signature from propulsion and airframe noise sources and to propagate the total noise to arbitrary ground observers. A basic summary of the original capability of ANOPP can be found in the

user manual (ref. 27). The description of noise prediction algorithms can be found in the theoretical manual (ref. 28). The most recent version of the code (i.e., ANOPP level 26) includes improved prediction methods that are needed for supersonic concepts. ANOPP was wrapped as part of the CDS project but was not included in the system-level process model. For the Gen 0 deliverable, the ModelCenter wrapper was updated to include all of the ANOPP L26 components.

The community noise assembly includes a number of subassemblies and components (see Figure 22). Each component writes a section of the input file, and the *joinFiles* utility component combines them into a complete input file. The execute component then runs ANOPP, and the parse component reads data from the ANOPP output file. The noise metric is effective perceived noise level (EPNL) at the approach, takeoff, and sideline microphones that are used for certification. Only jet noise and fan noise sources are deemed necessary for the test cases that are described in section 5. However, all potential noise sources and noise metrics are available, and an alternate process could be implemented. That new process would still involve assembling an ANOPP input file, executing the input file, and collecting the predicted noise values.
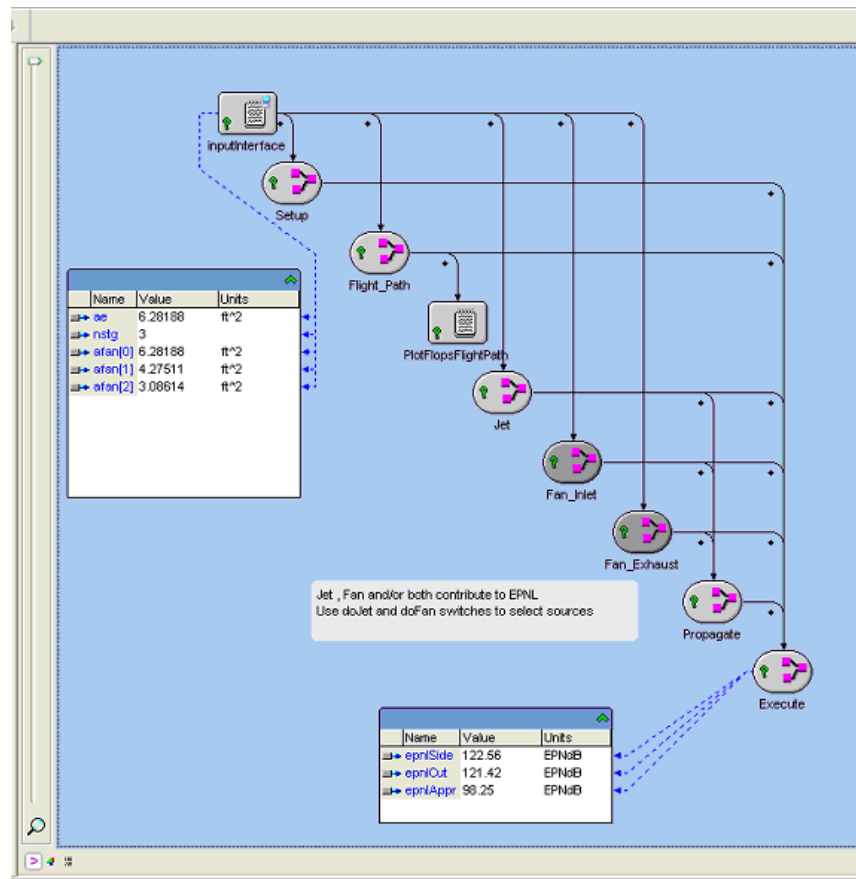

Figure 22. ANOPP implementation in ModelCenter.

The technical challenge that is posed by the ANOPP capability is that it must be customized for each new aircraft. Different sources of noise (e.g., jet, core, and airframe) will dominate the overall noise measures, depending on the type of engine and on the noise shielding that is provided based on nacelle location. The current plan is to link all possible noise sources and provide flags that will use ANOPP control language to skip those sources that are deemed unimportant. This strategy is being tested with jet, fan inlet, and fan exhaust sources to assess the

31

computational cost and ease of use. One downside to this approach is the large number of inputs from FLOPS and NPSS, which must be linked to the community noise assembly. If these links can be created easily using ModelCenter's automatic linking tool, then this strategy will be attractive.

4.9.4   Cost Analysis

The current MDAO framework includes the Aircraft Life Cycle Cost Analysis (ALCCA) code for calculating aircraft cost (ref. 29). ALCCA is a weight-based cost code that was wrapped and tested during the CDS project. The code itself required no modification to meet the needs of the Gen 0 framework. However, because the cost estimating relationships (CERs) used by ALCAA are predominantly weight-based, the results are uncertain as a result of the weight-estimating practices for the propulsion and airframe as cited in sections 4.2.2 and 4.6, respectively. The ALCCA code also requires information regarding manufacturing processes, material selection, economics, and various subjective complexity inputs, which may typically be unknown at the conceptual level and ill-defined in general.

The newly released Process-Based Economic Analysis Tool (PBEAT) developed by NASA, Boeing, Pratt & Whitney, and Rocketdyne is currently being exercised in support of the SUP, SFW, and Hypersonics Projects under the FAP, and will be in subsequent MDAO framework releases. PBEAT utilizes engineering and programmatic information (e.g., weights, dimensions, materials, platform descriptors, programmatic development phase descriptors, and technology readiness levels) to compute a base cost, then employs consistent engineering criteria to determine multiple cost complexity drivers that refine tens of thousands of implicit CERs, which are used to compute an estimate. The CERs are manufacturing-process-based as well as development-process-based. The operations and support capabilities are still evolving; however, the Tailored Cost Model (ref. 30) has been integrated as an initial means for computing life cycle cost and cash flow metrics. Most of the PBEAT inputs are hierarchical, enabling users to compute coarse estimates with limited information and subsequently refine these estimates as more information becomes available. Apart from computing an independent estimate, an analogy mode can also be employed for performing affordability trade studies or for populating unknown inputs by drawing information from benchmark estimates of existing systems. Deterministic and multiple stochastic methods are used within PBEAT for estimating cost-risk.

5   Assessment of the MDAO Framework

The Gen 0 framework was recently completed, and several design and assessment activities are underway. The business jet assessment shows the overall improvement in the systems analysis capabilities that are provided by the framework. The business jet that is modeled is used for a study of integrated engine, plume, and CFD analyses of a low-boom supersonic aircraft; this study was performed concurrently with the Gen 0 framework development. The Concorde assessment is used to demonstrate the improvement in the turnaround time for a single aircraft analysis that is enabled by the Gen 0 framework. In the final subsection, the strengths and weaknesses of the framework are discussed.

5.1   Supersonic Business Jet Design

As part of the validation and development of the Gen 0 framework, a supersonic business jet (SBJ) design was created. The relative benefits and penalties of airframe configuration and engine design changes were evaluated in terms of typical aircraft metrics such as ramp weight, fuel consumption, sonic boom loudness, and community noise. The airframe concept is similar to one that is described in reference 16. For this test case, an aircraft design and

analysis process has been developed using the majority of the available Gen 0 components and the Gen 0 framework.

The actual SBJ design was initiated prior to the Gen 0 framework availability but was conducted concurrently with the low-boom design subprocess development. With the completion of the Gen 0 framework, the SBJ design has now been modeled in the framework and can be redesigned with the more powerful integrated tools that were not available during the initial design process. Additionally, as missing pieces of the process that was used for the initial design are identified, those tools (or new tools that are developed to execute that function) will be added to the overall SUP system. For instance, the determination of fuel tank locations and volumes is not part of the current system; these calculations were done by hand for the original SBJ design. A desirable improvement in the system would include an automated process for defining gross locations for fuel tanks (e.g., a fuselage tank can be located in feet from the back of the passenger cabin, or wing tanks can be located in defined sections of the wing). Tanks could quickly be created in VSP, and simple rules could be established for automatically updating them with geometry changes.

5.1.1   Model Objectives

The basic objective of the SBJ analysis model was to design a configuration for reduced sonic boom and increased cruise efficiency and to show the impact of the engine plume, for over- and underexpanded nozzles, on the sonic boom ground signature with a multifidelity approach. A number of different aspects were involved in meeting the objective. A fuselage-shaping optimization process was tested by using low-fidelity tools. A low-fidelity plume model was developed for plume-shape prediction, based on NPSS engine data and nacelle geometry. A medium-fidelity analysis process for sonic boom loudness was developed. A key input to the high-fidelity sonic boom analysis for the study was the *dp/p* signature at six body lengths from the vehicle. Although the CFD analysis was performed outside the framework, the remainder of the analysis (e.g., conceptual geometry, grid generation, plume prediction) was automated, which eliminated a significant amount of manual data processing that was necessary previously.

5.1.2   Model Development

A low-boom configuration with a gross takeoff weight of 100,000 lb was used as a baseline in a study of integrated engine, plume, and CFD analyses of a low-boom supersonic aircraft. 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 reference 16 for more details regarding the development of this configuration. The three-view of this configuration is reproduced in figure 23.
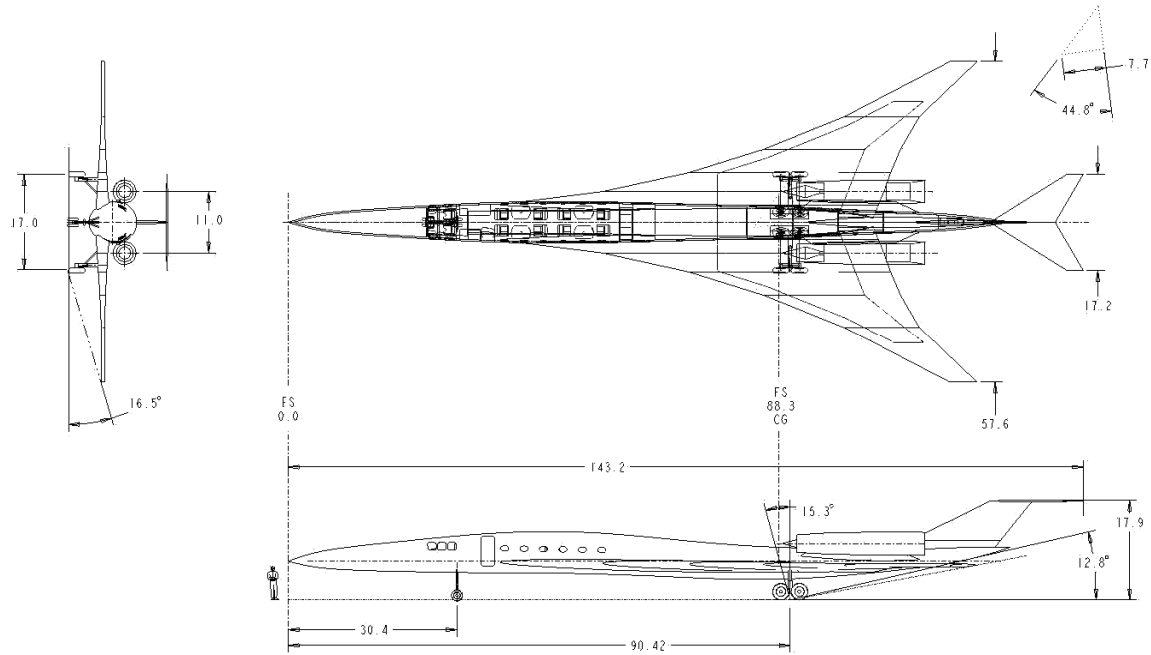
Figure 23. Three-view of low-boom baseline configuration.

The engine cycle used for this study is an MFTF. The maximum combustor exit temperature is 3,500 °R, and the high pressure compressor (HPC) discharge temperature is limited to 1,700 °R; these values are representative of technology levels from the NASA High-Speed Research Program. 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 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 aerodynamic design point (ADP) of Mach 1.8 and altitude of 52,000 ft. Given these constraints, the fan and HPC pressure ratio and the throttle ratio are optimized to minimize the cruise SFC. Previously, a baseline propulsion system would have simply been provided, or in rare cases, one would be selected from a few choices by individually flying them on the baseline aircraft. The results for the engine that was used in this study are shown in table 4. The flow path and weight results from WATE++ are shown in figure 24 and table 5. Note that during the study the ADP was changed from 55,000 to 52,000 ft, and the propulsion system was re-optimized; this update was possible as a result of the existing framework.

Table 4. 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

34
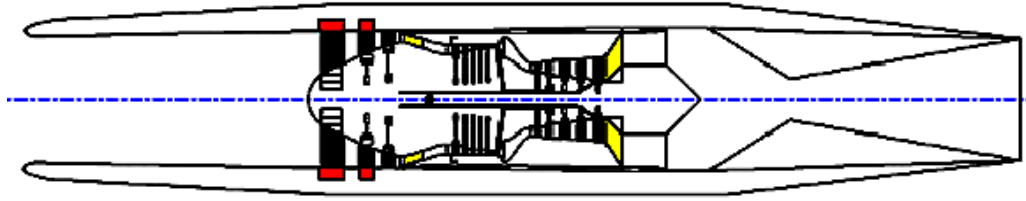
Figure 24. WATE++ flowpath.

Table 5. WATE++ Analysis Results

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

Also note that the MFTF optimization is decoupled from the mission, noise, and takeoff-landing analyses because the primary focus in this study is on the low-boom and low-drag properties of the concept. Therefore, the engine that was generated by the current optimization process is not necessarily the most desirable engine at the system level. For example, airport noise caused by the current engine might exceed the regulatory limit. However, the framework did enable the inclusion of the weight and geometry for a mixer ejector nozzle as a separate component, which allowed for a more reasonable mission performance analysis with at least a semblance of meeting community noise restrictions.

One of the key technical goals was to create a low-fidelity plume-shape-prediction method that is robust, efficient, and accurate. The method described in section 4.2.4 uses the geometric and thermodynamic cycle data generated by the NPSS/WATE++ component to construct nacelle and plume boundary OML's to be used for low-fidelity aerodynamic and boom analyses. Then, the high-fidelity and experimental results will be used to calibrate the low-fidelity plume method. These goals are quite challenging; however, an even bigger challenge is to develop a subprocess that operates with minimal human intervention.

The technical goals were accomplished by using smoothing techniques to create a nacelle shape that was based on a few engine diameters and segment lengths reported by the NPSS/WATE++ component (see section 4.2.3). Figure 25 shows the low-fidelity plume shape, illustrated by the solid black line, superimposed over a CFD plot of Mach number contours and stream traces for an underexpanded nozzle. Figure 26 illustrates the plume shape behind a typical supersonic business jet. The extent of the plume is indicated by the color pressure contours at a vertical cut behind the aircraft. This information is important to supersonic design because of its impact on the aft portion of the sonic boom signature as a result of both the initial shock caused by plume interaction with the freestream and the downstream nozzle shock train propagation beyond the plume boundary. If the underexpanded nozzle performance is exploited to potentially reduce sonic boom, there can be an adverse effect on thrust, SFC, and aerodynamic performance, which must be addressed in the overall MDAO process.

---

[*] Bare engine weight is WATE++ "bare engine weight" minus "engine mount weight" minus "fan exhaust cowl weight" (tailpipe) weight minus nozzle weight.
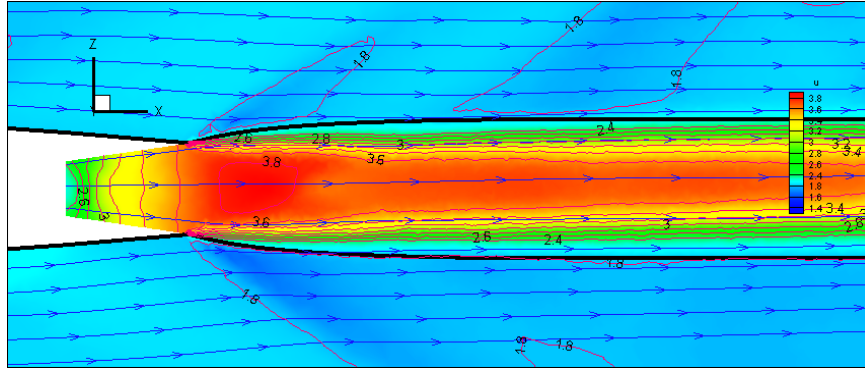
Figure 25. Comparison of low-fidelity plume shape with CFD results.
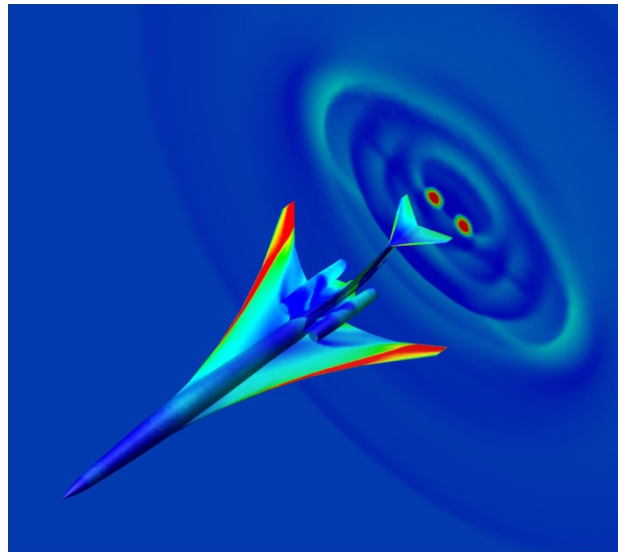

Figure 26. CFD results with plume shape indicated.

The original nacelle that is shown in Figure 23 is automatically replaced with the nacelle that is generated by using NPSS/WATE++ (Figure 4) and the fuselage is reshaped for a low-boom signature. The engine plume shape is varied to explore the possibility of favorably shaping the aft part of the ground signature. The plume shape is varied by changing the ratio of the ideal nozzle area ratio (exit area/throat area) to the actual nozzle area ratio (K_{noz}) in the propulsion component of the current ModelCenter process. 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 overexpanded and underexpanded nozzles, respectively. A few trials of varying K_{noz}, quickly demonstrated that only underexpanded plumes had any affect 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 affect on the ground signature, as predicted by PBOOM (see Figure 27). For this underexpanded case, K_{noz} has a value of 1.1. For the second case, a fully expanded nozzle was used to create a plume that looked like a cylinder of the same radius as the nozzle exit. Using the low-fidelity plume model, the K_{noz} parameter was adjusted until a cylindrical plume shape was achieved, which resulted in a K_{noz} value of 0.96. For the third case, because overexpansion of the nozzle had no detectable affect on the ground signature, a plume that was only slightly overexpanded was chosen, 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 27, Figure 28, and Figure 29. Note that the labels "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

36

is actually very slightly overexpanded. As a result of the relatively large nozzle internal divergence angle, the low-fidelity model requires a slightly overexpanded nozzle to achieve a cylindrical plume shape.
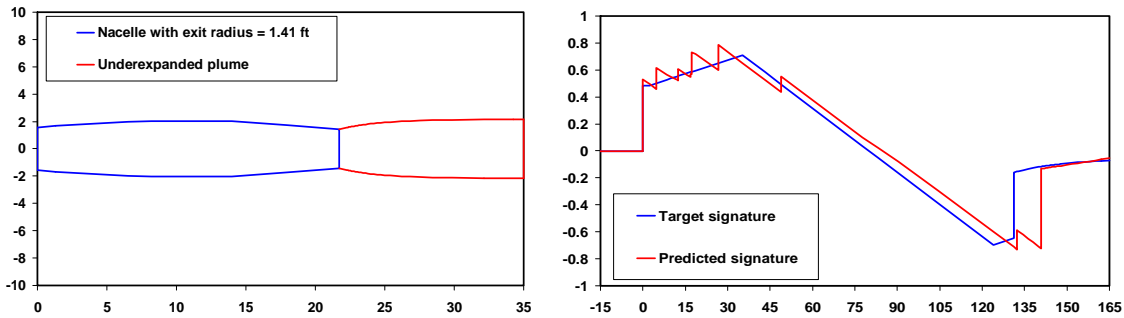

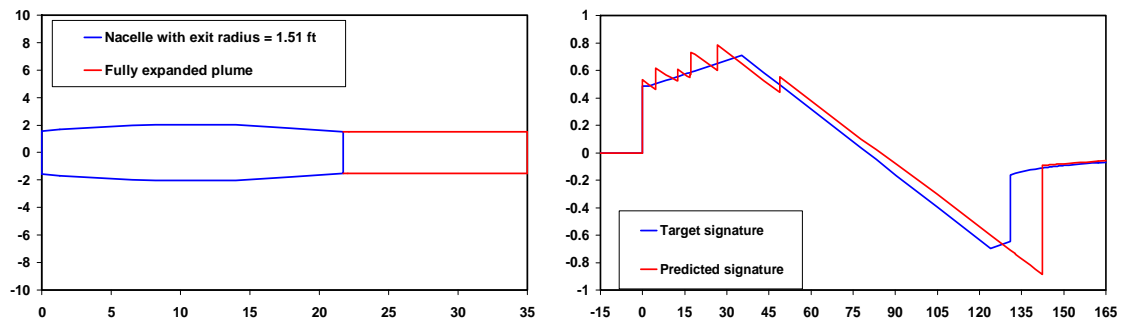Figure 27. Underexpanded nozzle and predicted ground signature by PBOOM.

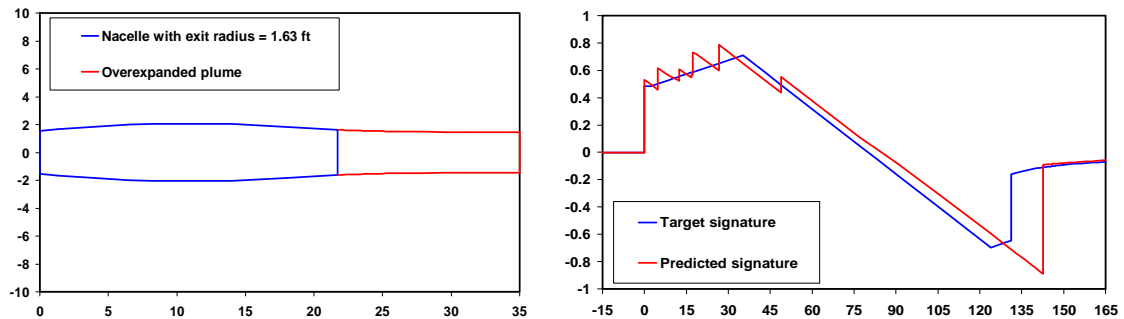Figure 28. Fully expanded nozzle and predicted ground signature by PBOOM.

Figure 29. 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, because the nacelle that was generated by the heuristic process described in subsection 4.2.3 is aerodynamically efficient, the propulsion system design parameters are fixed, and the perturbations to the nozzle exit area are small; thus, no redesign of the nacelle OML was required. Because the framework only executes those components that are required, the elapsed time from setting K_{noz} to obtaining the sonic boom ground signature that is shown in figures 27-29 is less than 1 minute, or just over 1 minute if the fuselage is reshaped. Once the desired low-fidelity ground signature is generated, the process to generate the watertight geometry in VGRID format (Figure 30) takes an additional minute.
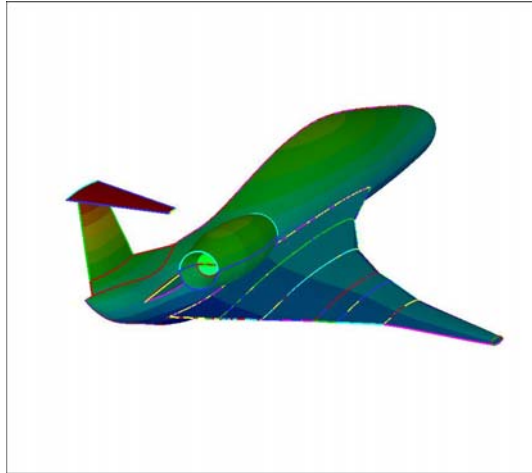
Figure 30. Supersonic business jet in VGRID format.

The CFD analysis process is to run AutoSrc, VGRID, SSGRID, and USM3D by using standard scripts with default inputs. The purpose of having CFD specialists run the CFD analyses is to fine-tune the default inputs and generate intermediate solution files to assist in the debugging of the Gen 0 framework. As a result, the actual turnaround time from submitting the input files and data (i.e., geometry and NPSS boundary conditions) to receiving all the USM3D solution files was 18 days. This highlights the importance of automating the CFD analysis in future releases of the MDAO framework. To obtain a complete mission analysis requires two additional minutes to update the aerodynamic data and compute community noise. The entire process requires approximately 8 minutes from generating the propulsion system data for mission, noise, and geometry until obtaining the sonic boom ground signature, range, community noise, and loudness estimates.

5.1.3 *Analysis Model Benefits*

Using the MDAO Gen 0 system for the SBJ study resulted in several benefits: automated transfer of data, reduced turnaround time, consistent analysis, and rapid reanalysis.

The automated transfer of data between the steps in the analysis process provided a significant reduction in the time that was required for analysis. One example is the propulsion system optimization process. The previous manual process required a systems analyst at NASA Glenn Research Center to generate the baseline engine deck, weight, and flow path data, based on an initial guess of thrust and thrust lapse. This information was emailed to a systems analyst at NASA Langley Research Center, who then manually generated a nacelle OML, generated the aerodynamic data for mission analysis, and ran FLOPS to obtain an updated thrust and thrust lapse. Frequently, the propulsion system optimization process would end at this point. If time permitted, the process might be repeated to obtain results at three different bypass ratios, and the propulsion system that produced the greatest range would be considered the optimal system. With the Gen 0 SBJ analysis model, the optimization process is reduced to a single click of a button to obtain a good starting point, as described in the previous subsection. The new capability provides the option to perform an overall systems design by using the DOE capabilities of ModelCenter, as illustrated in Figure 31.
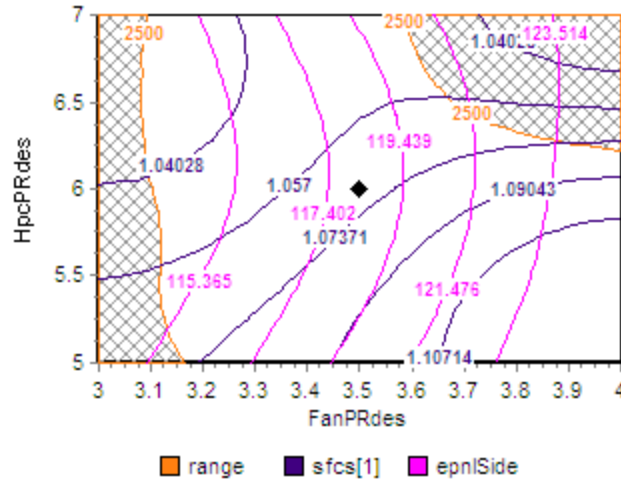
Figure 31. Sample contour plot with component pressure ratios as design variables.

The basic "toolbox" approach for the MDAO Gen 0 system provides the flexibility to use only those analysis components that are necessary for the problem at hand. For example, one of the Gen 0 components that did not need to be reanalyzed during the fuselage shaping was NPSS. Because the propulsion design assembly is segregated into various components based on discipline, NPSS can be used to provide an increase in the aerodynamic analysis fidelity by changing the size of the nacelles and the shape of the plume. This application of NPSS to optimize fuselage shape does not require that the time-consuming NPSS components that generate the data for mission and community noise analyses be run again.

5.1.4   *Potential Improvements*

The Gen 0 SBJ analysis model was developed prior to completion of all of the desired tool development activities. Given the sonic boom focus of the study, the utility of the model will be greatly enhanced once the high-fidelity aerodynamics and PCBOOM components have been fully debugged and automated. Having PCBOOM as an integral part of the analysis process (as opposed to performing the analysis separately) will enable a high-fidelity assessment to occur during the low-boom shaping rather than after it is completed. Improvement of the Gen 0 MaSCoT tool would enhance the stability and control analysis, which currently includes a number of simplifying assumptions.

Although the integrated process greatly reduces the overall analysis time, some "bottlenecks" still limit the execution speed. For example, the current VSP plug-in is very slow. Work currently is being performed by Phoenix Integration under a SFW NASA Research Announcement (NRA) to develop a new VSP plug-in that should speed up the execution. Another concern with the current capability is the amount of time that is required to initially build a new process model. A tradeoff certainly is made between the flexibility that is offered by the modular, "toolbox" approach of Gen 0 and the time that is required to build a customized model. As more Gen 0 models are developed and a repository of complete models becomes available, these models hopefully can serve as starting points to reduce the time that is required to develop each new model.

5.2   Concorde Assessment

In 2007, the SUP project carried out a number of benchmarking activities in various disciplines. In the area of MDAO, the goal of the benchmark exercise was to quantify the time and accuracy involved in modeling an existing vehicle from limited knowledge. The Concorde aircraft was selected as the vehicle for this benchmark activity.. The

study was repeated with the current SUP MDAO system capability to measure the improved cycle time and accuracy of the newly developed integrated analysis process. These efforts are summarized in the following subsections.

5.2.1   Results: CDS framework

The cycle time from the initial benchmarking with the CDS generation tools is shown in Table 6. The individuals who worked on the subtasks shown in the table kept track of their wall clock time on each activity. The times shown include the actual time spent on these tasks and do not include time spent in meetings. The tools that were integrated into the ModelCenter environment during CDS were used and these had already greatly reduced the time that was required to make multiple runs once the initial set up was completed. In table 6, the "preliminary research and data gathering" task included gathering the initial data that were necessary to develop the propulsion and airframe models, including from library and online sources. For the "initial geometry development" task, the time included only the time that was spent developing the initial VSP geometry, as illustrated in Figure 32.

Table 6. Cycle Time for Concorde Benchmark Study

| Task Description	Time, hrs
Preliminary research and data gathering	9
Propulsion system cycle analysis (NPSS)	16
Propulsion system weight and flow path analysis (WATE++)	21
Emissions calculations	10
Initial geometry development	6
Aerodynamics	2
Takeoff and landing aerodynamics	10
Remaining FLOPS setup/input	4
Setting up the complete ModelCenter model	9
Exercising the ModelCenter model and resolving issues	8
Mission and weights analysis	2
Generating NPSS/WATE data for noise analysis	8
Generating takeoff and landing trajectories for noise analysis	11
ANOPP noise analysis and ST2JET comparisons	32
Total analysis	148
Documentation (includes additional research for more detailed data)	120
Total	268
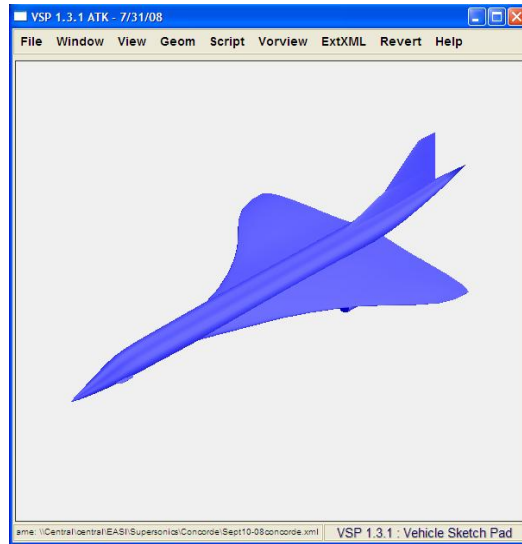
Figure 32. Concorde model in VSP.

In general, the geometry, aerodynamics, performance and sizing tools were already fairly well integrated in the process that was used for this analysis. As expected, the longer cycle time items were the propulsion system development, the noise analysis, and the exchange of needed information. For a single forward flowing analysis process, the integration was handled by passing FLOPS-compatible engine decks and FLOPS-generated takeoff performance runs between the responsible parties at NASA Langley Research Center and NASA Glenn Research Center. Additional integration of low-speed aerodynamics was required.

The level of detail in the analysis was typical for quick turnaround analyses. Many discipline areas were not included in the analysis because of the time constraints that were imposed by the quick turnaround requirement. Instead, engineering judgment was used to account for those areas. As a result, the unmeasured error in the system may be large. When better tools can be incorporated for timely analysis, the discipline areas that were not evaluated can be included in the process. The study compared various discipline and system-level metrics; while some showed wide variation relative to the published data, the overall model did a fairly good job of predicting and modeling the Concorde. The specific results are not given here but are expected to be published in the future.

5.2.2   Results: Gen 0

For the current Gen 0 Concorde analysis, the entire analysis process was completed within the SUP MDAO framework. While some of the parts of the original analysis were not redone, for example, the propulsion system development, the time savings for developing the overall Concorde model was significantly reduced. The baseline SUP framework was developed around a business-jet-sized aircraft with significant differences in the modeled aircraft components. Thus, model modifications were necessary because of the number of engines, the number and locations of control surfaces, and other differences (e.g., compare figure 7 with figure 32).

Because the overall SUP MDAO framework was only recently completed when the analysis was initiated, several bugs in the overall framework were discovered and corrected. Correcting these bugs was not included in the cycle time, whereas, the actual Concorde model-specific debugging was included. The included time would be tool-specific input-value debugging that would typically involve initializing various code inputs.

The overall cycle time that was required with the newly integrated system was less than 24 hrs of wall clock time. Analysis results were compared with the initial study results to verify that the integrated process did at least as well

as the original benchmarking. Because most of the same analysis tools that were used the first time were used again in this analysis but were fully integrated, the results should have the same level of uncertainty. Many of the new modules were not used because the analysis did not call for them, and validation data were not available in most cases. The main objective between the two studies was the time savings to be realized with the fully integrated tool set, and that improvement was fully realized.

5.2.3   Strengths and Weaknesses: Gen 0

Although this was not an exhaustive study, it was instructional to see how time in the process was and is now being spent. The increased productivity cannot be fully realized until the model is run iteratively for trade studies, sensitivity studies, or general debugging. The integration of all of the components significantly reduces the data consistency errors that can arise when codes are re-executed to correct one problem but the new resulting data is not updated properly throughout the model. To illustrate the benefits of the new system, a trade study was run to examine how sensitive the system-level results were to the potential modeling error in T_4 in the NPSS model. This typical trade study was completed in a few minutes. Shown below are some sample results from the Concorde model. For the study, the max thrust with afterburner was held constant and the resulting impact of T_4 on cruise SFC, range, engine weight and cutback noise was examined. The results are shown in Figure 33. The ability to quickly look at the sensitivity of various responses to an uncertain input parameter can be of significant help to the analyst in deciding how much effort or additional fidelity is required when modeling a low-level discipline parameter to assess system-level metrics.
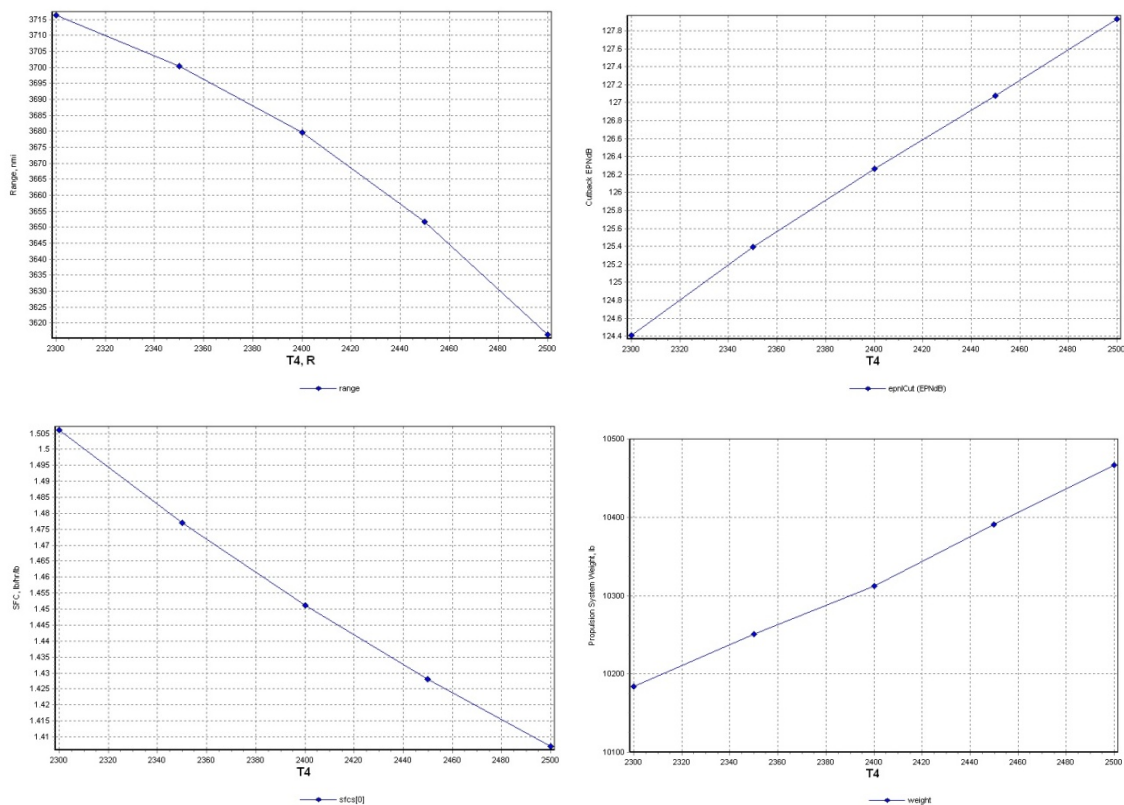

Figure 33. Sensitivity to T_4 modeling.

5.3 Overall SUP MDAO Framework Strengths and Weaknesses

In this section, the overall strengths and weaknesses of the SUP MDAO framework are discussed. Because this is the first phase of the MDAO system, which is planned for completion in 2016, the lessons learned here will be used to guide future development.As explained in prior sections, the SUP components, processes, and test cases were developed and tested within an existing commercial framework called ModelCenter. Although the ModelCenter framework is itself a separate software capability, the MDAO system and the incorporated methods are intimately linked to this framework. While many of the capabilities that have been previously discussed could be migrated to an alternate framework in the future, the existing framework has many features that are inherently important to the overall capability.

5.3.1 MDAO Framework Strengths

The ModelCenter framework provides systems analysts with several essential utilities. As discussed previously, to design a supersonic vehicle the designer must know how changes in the engine and the airframe affect sonic boom, community noise, and fuel efficiency. The DOE tools and visualization tools that are provided by the ModelCenter framework meet this need. All of these tools were improved in version 7 of the ModelCenter software release. The term DOE refers to a set of statistical tools that can be used to investigate the effects of multiple input variables on multiple output responses. The ModelCenter DOE tool contains sampling methods, such as Full Factorial, Central Composite, and Latin-Hypercube methods. In the experiments performed to date with the supersonic process model, the Latin-Hypercube method has proved to be a good choice of sampling method: the sample points are selected randomly with equal probability from any part of the design space; mathematically, this gives a clearer picture of the nonlinear trends in the data. As a practical matter, the random sampling is favored because the bounds on a design variable are often not well-known and the analysis can return invalid results for untested combinations of design variables.

The new visualization tools are linked to the DOE tools and provide an efficient means to inspect n-dimensional data spaces, where n is quite large. The ModelCenter Data Visualizer includes seven plot types for viewing data from a DOE: one- and two-dimensional histograms, a parallel coordinates plot, a scatter plot, a scatter matrix, and two types of glyph plots. Parallel coordinates plots and glyph plots facilitate the viewing of more than three dimensions at a time through factors such as glyph size and color. The n-dimensional plotting allows a systems analyst to discover how an entire set of variables must be varied in a coordinated manner in order to arrive at the best designs.

Scatter plots are particularly useful for capturing multiobjective values on a single plot. For example, Figure 34 is a plot of noise at one of the sideline certification points as a function of fan pressure ratio (i.e., the EPNL that was predicted by ANOPP is plotted against the fanPR from NPSS). This plot was produced by a DOE with six variables: fan pressure ratio, top of climb (TOC) thrust, HPC pressure ratio, outer-section wing sweep, and two flap hinge locations. This plot clearly indicates that community noise increases with fan pressure ratio. Similar plots of each variable indicate that the other five variables have a secondary effect on community noise. The preference shading in Figure 34 is used to capture another response—range—on the same graph as community noise. Red dots indicate low range, and blue dots indicate high range. In this example, low fan pressure ratio has a detrimental effect on the range as predicted by FLOPS. Figure 35 is an alternate way to look at this same data; the maximum sideline EPNL is plotted against both fan pressure ratio and TOC thrust. Also shown is an engine scaling factor—ESF—which will equal unity whenever the mission analysis code uses the results of the engine cycle analysis without scaling. This plot reveals that the noise estimates at the lowest fan pressure ratio could have a higher uncertainty because they depend on scaled engine data.

Plots such as those shown in Figure 34 and Figure 35 can be used to make design decisions but also can uncover modeling errors. For example, if the scatter plot does not agree with a known trend (e.g., if fuel use does not trend appropriately with engine thrust), then a plot such as the one shown in Figure 34 will encourage a careful check of the modeling assumptions. Similarly, contour plots such as the one shown in Figure 35 can indicate a problem if responses that are known to be linear show a large amount of nonlinearity.
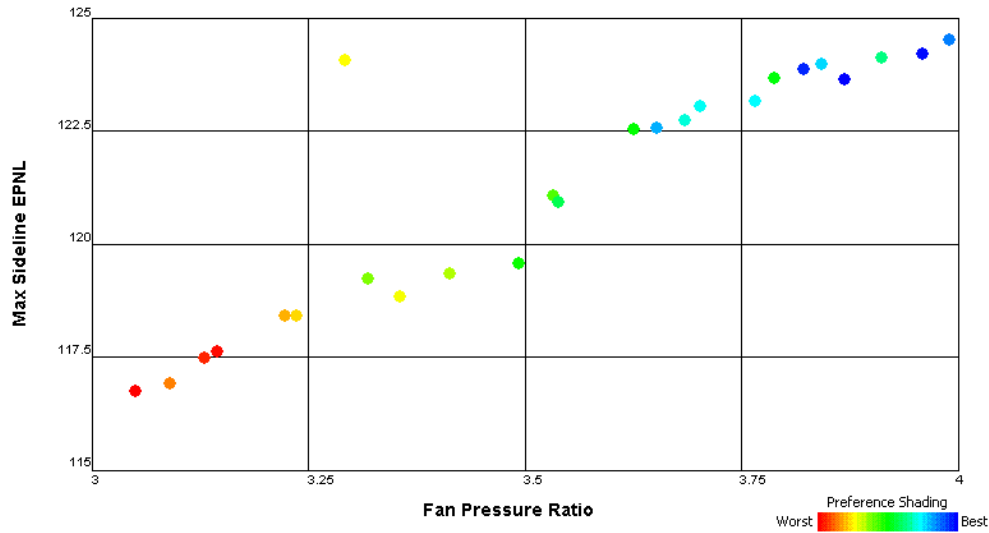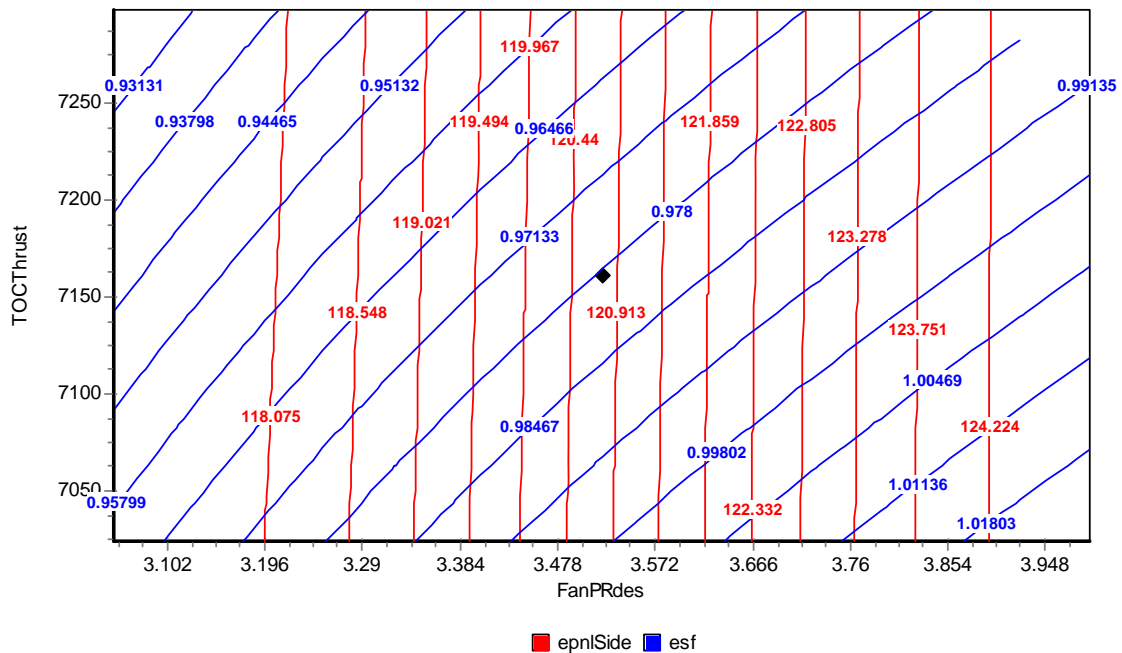

Figure 34. Typical supersonic design results.

Figure 35. Typical contour plot; Red lines are EPNL, and blue lines are ESF.

Figure 36 is another scatter plot that is based on the same six variables that were used for Figure 34. In figure 36, the independent variable is the outer-section wing sweep. This variable was created in the ModelCenter framework by

linking several other variables that define the sweep of individual wing sections. The scatter plot shows the effect of outer-section wing sweep on the balanced field length as predicted by FLOPS. The preference shading is used to indicate sonic boom loudness on the same graph. Figure 37 and Figure 38 can be compared with Figure 7 to visualize the changes in wing geometry that occur with a change in the outer-section sweep from 60 to 70 deg. These geometry plots are generated automatically whenever the user activates the associated plot options. Because of the versatile plotting capability and the simple design-variable linking capability, the addition of a new wing sweep variable requires just a few minutes. After the new variable was added, the DOE tool required approximately 4 hrs to run 30 cases and to produce the scatter plots. While the outboard wing sweeps shown in Figure 37 and Figure 38 are likely to be unreasonable, the actual penalties that result from the high outboard sweep are captured by other responses in the DOE output. The visualization capabilities make it easy to see where the DOE design variables transition from reasonable values to undesirable regions of the design space. This immediate visual feedback to the user dramatically enhances productivity.
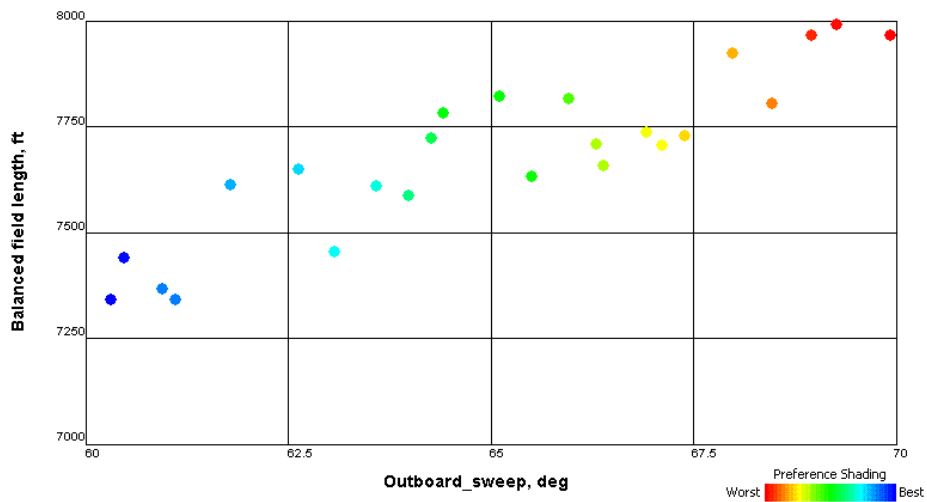

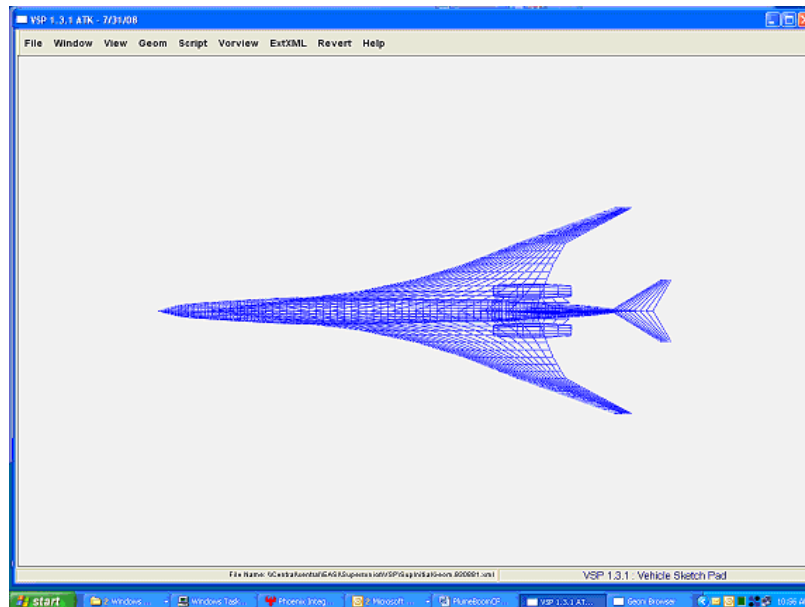Figure 36. Preliminary results showing effect of outboard sweep angle.

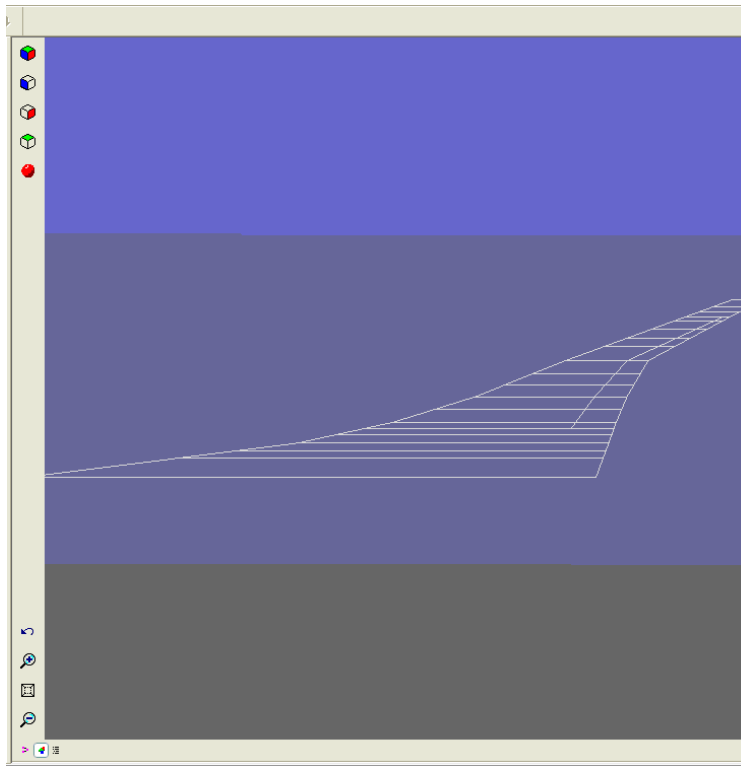

Figure 37. Wire-frame plot illustates a high sweep angle.

45

Figure 38. Geometry viewer reveals wing sections linked to outer wing sweep variable.

5.3.2 MDAO Framework Weaknesses

Several weaknesses in the ModelCenter framework and in the MDAO implementation were discovered during the assessment process. These weaknesses involve the archiving of models and trade studies, the installation of framework and models, the customization of models and the execution of DOE runs. Many of the causes for these weaknesses have been diagnosed, and plans have been made to address them in the Gen 1 framework.

Archiving and reuse of models is a central goal of the MDAO framework milestone. But the Concorde study uncovered some weaknesses in that area. Older models, such as the 2007 Concorde study model, may refer to obsolete computer addresses or older versions of the software codes. That means that the model will not execute until all of the old references are updated. A model such as the one pictured in Figure 1 has over 200 components. Even though the modifications that are needed by each component are easy to accomplish, the job can be quite tedious. In the future, models and components need to be constructed with archiving in mind. In addition, each model should be saved with the associated baseline input and output files so that it can be tested for proper execution against a set of known results.

Installation proved to be another tedious job. Component plug-ins, such as Mathworks® Matlab®, Microsoft® Excel®, and NPSS, run on the local personal computer and must be installed and tested for each new user. This time-consuming job can be streamlined by having a folder on a central server that contains all such components along with installation notes and FAQ files.

Allowance for individual customization of components was another issue that became critical as more users wanted access to the framework. For example, many components have user-specified input files and optional output graphics. The component must include an easy way to activate these options and to change the names of input files or the paths to local graphics executables.

46

An additional weakness was uncovered while producing the DOE results presented above. By its nature, a DOE contains combinations of variable values that may not be physically realizable or that may be outside the range for which the component has been tested. Components should establish upper and lower bounds on inputs that have known acceptable ranges. Components also must fail immediately if some bound is exceeded or if some input file is missing, rather than waste additional computer resources. Components also must fail gracefully and must provide useful error messages.

6 Concluding Remarks

The supersonics MDAO team has successfully completed the SUP milestone to provide an initial supersonics MDAO capability. The current system includes all of the capabilities that were initially identified for inclusion in this release of the system. Two demonstration cases have been completed to show the improved capability, as well as demonstrate a less than two-day cycle time to model a well-characterized concept. The current system has been shown to greatly increase the design and analysis speed and capability, and many future areas for development were identified. This work has established a state-of-the-art capability for immediate use by supersonic concept designers and systems analysts at NASA, while also providing a strong base to build upon for future releases as more multifidelity capabilities are developed and integrated.

7 References

1. McCullers, Linwood A.: "The Flight Optimization System, FLOPS Users Manual," Updated April 2008 (Available with public distribution of software).
2. Phoenix Integration, Inc. "ModelCenter, Design Integration Software (Version 7.1)," Blacksburg, VA, 2007.
3. 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-2000-209795, August 2000.
4. Onat, E. and Klees, G. W.: "A Method to Estimate Weight and Dimensions of Large and Small Gas Turbine Engines," NASA CR-159481, January 1979.
5. Nash, Kyle L.; Whitaker, Kevin W.; and Freeman, Michael: "Predicting Exhaust Plume Boundaries with Supersonic External Flows," *J. of Spacecraft and Rockets*, Vol. 31, No. 5, Sept-Oct 1994.
6. Barger, R. L. and Melson, N. D.: "Comparison of Jet Plume Shape Predictions and Plume Influence on Sonic Boom Signature," NASA TP-3172, Mar 1992.
7. VGRID Unstructured Grid Generator, http://tetruss.larc.nasa.gov/vgrid/, accessed August 18, 2008.
8. Harris, R. V., Jr.: "An Analysis and Correlation of Aircraft Wave Drag," NASA TM X-947, March 1964.
9. Carlson, H. W.; Chu, J.; Ozoroski, L. P.; and McCullers, L. A.: "Guide to AERO2S and WINGDES Computer Codes for Prediction and Minimization of Drag Due to Lift," NASA TP-3637, November 1997.
10. Carlson, H. W.; McElroy, M. O.; Lessard, W. B.; and McCullers, L. A.: "Improved Method for Prediction of Attainable Wing Leading-Edge Thrust," NASA-TP-3557, April 1996.
11. 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.
12. Mack, R. and Haglund, G.: "A Practical Low-Boom Overpressure Signature Based on Minimum Sonic Boom Theory," *High-Speed Research: Sonic Boom Workshop*, NASA CP-3173, Vol. 2, 1992, pp. 15–29.
13. Seebass, R. and George, A. R.: "Sonic-Boom Minimization," *J. of Acoustical Society of America*, Vol. 51, No. 2, 1972, pp. 686–694.
14. Darden, C. M.: "Minimization of Sonic-Boom Parameters in Real and Isothermal Atmospheres," NASA TN D-7842, March 1975.
15. Coen, P.: "Development of a Computer Technique for the Prediction of Transport Aircraft Flight Profile Sonic Boom Signatures," M.S. Thesis, School of Engineering and Applied Science, George Washington Univ., Washington, DC, March 1999.
16. Li, W.; Shields, E.; and Le, D.: "Interactive Inverse Design Optimization of Fuselage Shape for Low-Boom Supersonic Concepts," AIAA-2008-136, January 2008.

17. McCullers, L. A.: FLOPS Weight Module Documentation. FLOPS Users Manual, Updated April, 2008 (Available with public distribution of software).
18. McCullers, L. A. and Lynch, R. W.: "Composite Wing Design for Aeroelastic Tailoring Requirements," *Air Force Conference on Fibrous Composites in Flight Vehicle Design*, Dayton, Ohio, September 1972.
19. Ardema, M. D.: "Analytical Fuselage and Wing Weight Estimation of Transport Aircraft," NASA TM-110392, May, 1996.
20. McCullers, Linwood A.: FLOPS Performance Module Documentation. FLOPS Users Manual, Updated April, 2008 (Available with public distribution of software).
21. McCullers, Linwood A.: FLOPS Takeoff and Landing Module Documentation. FLOPS Users Manual, Updated April, 2008 (Available with public distribution of software).
22. Crespo, L. G. and Kenny, S. P.: "Matlab Stability and Control Toolbox: Trim and Static Stability Module," NASA TM-2006-214536, November 2006.
23. MDAO GEN 1 Team: "Generation 1 (GEN 1) Integrated Multidisciplinary Toolset," *SFW.01.01.009 Milestone Report*, June 30, 2008.
24. Plotkin, K. and Page, J.: "Extention of PCBoom to Over-the-top Booms, Ellipsoidal Earth, and Full 3-D Ray Tracing," AIAA-2007-3677, June 2007.
25. Shepherd, K. and Sullivan, B.: "Loudness Calculation Procedure Applied to Shaped Sonic Booms," NASA TP-3134, 1991.
26. Stevens, S. S.: "Perceived Level of Noise by Mark VII and Decibels," *JASA,* Vol. 51, No. 2(2), 1972, pp. 575–593.
27. Gillian, R. E.: "Aircraft Noise Prediction Program User's Manual," NASA TM-84486, January 1982.
28. Zorumski, W. E.: "Aircraft Noise Prediction Program Theoretical Manual, Parts 1 and 2," NASA TM-83199-PT-1 and PT-2, February 1982.
29. Mavris, D.N. and Galloway, T.: "Aircraft Life Cycle Cost Analysis Version 6.0 User's Manual," Aerospace Systems Design Laboratory, Georgia Institute of Technology, Atlanta, September 2001.
30. Spitz, W.; Berardino, F.; and Johnson, J.: "Development Cycle Time Simulation for Civil Aircraft," NASA CR-2001-210658, January, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-06-2010	Technical Memorandum	

4. TITLE AND SUBTITLE

Initial Multidisciplinary Design and Analysis Framework

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ozoroski, L. P.; Geiselhart, K. A.; Padula, S. L.; Li, W.; Olson, E. D.; Campbell, R. L.; Shields, E.W.; Berton, J. J.; Gray, J. S.; Jones, S. M.; Naiman, C. G.; Seidel, J. A.; Moore, K. T.; Naylor, B. A.; Townsend, S.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

984754.02.07.07.12.04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-19768

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2010-216711

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 66
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Within the Supersonics (SUP) Project of the Fundamental Aeronautics Program (FAP), an initial multidisciplinary design & analysis framework has been developed. A set of low- and intermediate-fidelity discipline design and analysis codes were integrated within a multidisciplinary design and analysis framework and demonstrated on two challenging test cases. The first test case demonstrates an initial capability to design for low boom and performance. The second test case demonstrates rapid assessment of a well-characterized design. The current system has been shown to greatly increase the design and analysis speed and capability, and many future areas for development were identified. This work has established a state-of-the-art capability for immediate use by supersonic concept designers and systems analysts at NASA, while also providing a strong base to build upon for future releases as more multifidelity capabilities are developed and integrated.

15. SUBJECT TERMS

Conceptual; Design; MDAO; Multidisciplinary; Supersonics; System; Multi-fidelity

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	53	STI Help Desk (email: help@sti.nasa.gov)

19a. NAME OF RESPONSIBLE PERSON
STI Help Desk (email: help@sti.nasa.gov)

19b. TELEPHONE NUMBER *(Include area code)*
(443) 757-5802

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18