13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference
13 - 15 September 2010, Fort Worth, Texas

AIAA 2010-9319

High Fidelity Multidisciplinary Optimization (HFMDO)

Katherine Alston, PMP[1], Steven Doyle[2], and Tyler Winter[3]
M4 Engineering, Inc., Long Beach, California, 90807

Hongman Kim, Ph.D.[4] and Scott Ragon, Ph.D.[5]
Phoenix Integration, Blacksburg, Virginia, 24060

The High Fidelity Multidisciplinary Optimization application delivers physics-based multidisciplinary analysis and optimization (MDAO) capabilities that are required to develop next generation supersonic aircraft. The software tools and approaches accurately model prediction of vehicle performance, interdisciplinary couplings, and system-level evaluation of the benefits and risks. M4 Engineering (experts in high fidelity MDAO processes) is working with Phoenix Integration (developer of the industry standard ModelCenter MDAO framework) to combine their specialties to deliver a modular design environment suitable to high fidelity analysis and design of coupled systems. The key elements of this toolset include an object-oriented integration framework, common objects and analysis modules based on custom data types. The HFMDO system utilizes Geometry Manipulation by Automatic Parameterization (GMAP) and RapidFEM for advanced parametric geometry and grid generation technology for aerodynamic and structural models. M4 Engineering and Phoenix Integration are developing the HFMDO System using multiple, incremental builds. This paper describes the capabilities of Build 3.1 of the HFMDO System, which includes modules for addressing the disciplines of Geometry, Aerodynamics, Propulsion, Structural, Stability & Control, Mission, and Noise. The Aerodynamics Module is enhanced to support multi-fidelity aerodynamics. Through the use of RapidFEM, the Structural Module now supports generating finite element models based on parametric, or smart, structure definitions.

Nomenclature

V	=	velocity at current Mach number and altitude
C_{Lmax}	=	Maximum lift coefficient
L/D	=	Lift/Drag
Isp	=	specific impulse
W	=	weight
T	=	thrust
TOGW	=	takeoff gross weight
$C_{L\alpha}$	=	Lift Curve Slope
C_{Do}	=	Zero-Lift Drag

I. Introduction

RECENTLY there have been significant efforts to bring physics-based models into the preliminary design phase of aerospace systems. Physics-based models allow for a higher fidelity analysis. An example of this is the Integrated Hypersonic Aeromechanics Tool (IHAT) developed by a team that included M4 Engineering[1]. However, these existing systems do not provide the capabilities required for designing the next generation air and space vehicles. The individual modules are implemented in a manner that prevents them from being easily reused as

1 R&D Project Manager, 4020 Long Beach Blvd, Long Beach, CA, 90807
2 Aerospace Engineer, 4020 Long Beach Blvd, Long Beach, CA, 90807
3 Aerospace Engineer, 4020 Long Beach Blvd, Long Beach, CA, 90807, Member AIAA
4 Senior Software Engineer, 1715 Pratt Drive, Suite 2000, Blacksburg, VA 24060, Senior Member AIAA
5 Director of Research, 1715 Pratt Drive, Suite 2000, Blacksburg, VA 24060, Member AIAA

configurations and the design problems change. Since the modules are tied together using scripting languages in a rather ad-hoc approach, it is difficult to understand and modify the workings of the system. A system should deliver a suite of capabilities that can be utilized as required depending on the configuration and the problem being solved. As a system becomes more modular, modules can be more easily swapped in and out based on the user's needs without excessive system redesign. A modular framework that is highly configurable (Figure 1) provides the required foundation to address varying fidelity level analysis and design problems, which are applicable to multiple configurations and incorporate physics-based models.

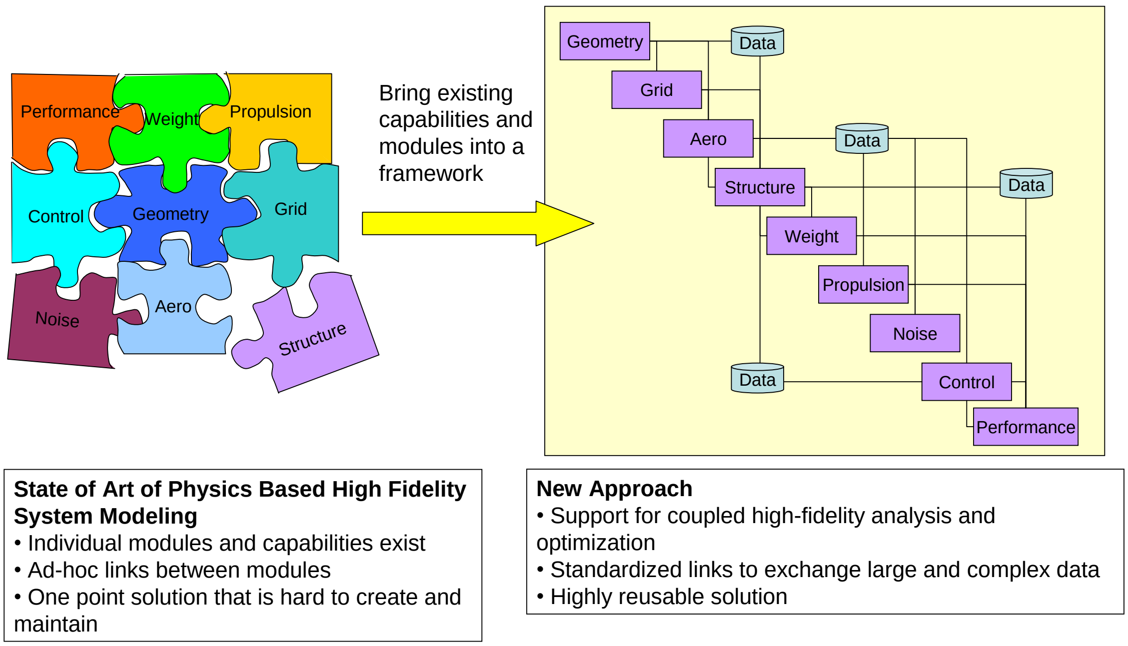

Figure 1. Create high fidelity, physics-based analysis and design capability that is modular and reusable.

Framework software tools (e.g., ModelCenter®[2]) standardize the common tasks of analysis code execution, job scheduling, the use of distributed computing resources, and the transfer of data from one analysis code to another. Unfortunately, framework tools have been less successful in high fidelity applications. This is because the development of high fidelity MDAO processes gets increasingly difficult as more and more complex data and information need to be exchanged. As an example, consider the interaction between aerodynamics and geometry. When the Geometry Module is executed, a Geometry Calculations Object is instantiated. Subsequently, this object is passed downstream to the Aerodynamics Module. This object contains various methods designed to calculate important geometric quantities (i.e. flap area, vertical tail span, etc.) by interrogating the baseline and/or morphed model for use in the Aerodynamics Module. In order to avoid duplication of work, it is critical for the framework tools to provide capabilities to easily transmit complex data between modules. Another capability that is key to the success of physics-based modeling is the accurate modeling of geometry and the automatic generation of quality grids and meshes. The analysis of coupled systems requires consistent conversion of geometry data to external grids and internal structural meshes. The automatic mesh generation should be flexible enough to handle large geometry variations and be adaptable to different configurations. Lastly, in order to perform meaningful trade studies and optimization in a reasonable amount of time, the analysis models should be updated in an automated and efficient manner as design variables are changed.

HFMDO is a multidisciplinary analysis and optimization toolset designed to address these issues for next generation vehicle applications. HFMDO develops an object-oriented integration framework that allows users to efficiently link high fidelity analysis modules. This framework significantly reduces the problem setup time by simplifying the definition of interdisciplinary coupling, allowing the creation of complex data objects, and eliminating laborious manual data conversion. HFMDO develops a library of common objects and analysis modules based on custom data types. The initial focus is on high fidelity aerodynamic and structural analysis disciplines and the associated objects (e.g., aerodynamic database). The HFMDO system succeeds in sharing complex data by utilizing an object oriented approach in which upstream modules create objects that are used by downstream modules upon demand. Both the data and the methods reside in the object and downstream modules request the data when needed. HFMDO implements automatic mesh generation and morphing through advanced parametric geometry and grid technology for multidisciplinary modeling[5]. M4 Engineering has developed a parametric grid morphing tool, Geometry Manipulation by Automatic Parameterization (GMAP)[6] and a parametric FEA model generator for internal structures (RapidFEM[7]). These tools are integrated into the framework environment so that engineers can quickly integrate FEA/CFD analyses, morph geometry, re-mesh, apply loads, and generate useful results. Through careful automation of the analysis process, the HFMDO system allows configurations to be rapidly assessed, allowing many variations to be considered in a relatively short time. This facilitates the implementation of numerical optimization techniques that can be used to help determine the optimum design. An example application demonstrates the use of this new MDAO framework and analysis modules for the high fidelity MDAO of a relevant supersonic fixed wing vehicle configuration. Additionally, two advanced capabilities have been implemented within the HFMDO system. The first capability includes a multi-fidelity aerodynamics tool that combines low fidelity and high fidelity aerodynamic results into a mid fidelity aerodynamic database. This new mid fidelity aerodynamic database contains "corrected" results across a large flight envelope. The second capability consists of a parametric meta-mesh enhancement implemented within RapidFEM. This parametric (or smart structure) utility allows the user to parametrically define the structural planform layout in terms of useful parameters.

II. HFMDO Development Approach

Developing high fidelity MDAO currently requires extremely sophisticated skills of both engineering and software development. The approach defined herein will help engineers focus on engineering problems instead of on programming issues. This process is leading to a paradigm shift in the way high fidelity MDAO is performed.

A. Incremental Build Approach

An incremental build approach is being used to develop this HFMDO system. The incremental build approach has been successful in implementing the philosophy of "Thinking big but starting small". The overall system is too complex to attempt to build in a single step. Smaller, more manageable development milestones have been established and risk has been greatly reduced. This paper specifically addresses Build 2, which culminated at the end of Year 2 with software and documentation being delivered October 31, 2009.

Build 1 established the development framework, integrated prototype modules and provided a proof-of-concept demonstration. This first build was comprised of the Geometry, Aerodynamics, Propulsion, and Structural Modules implemented in ModelCenter. ModelCenter provided the infrastructure for the object-oriented integration framework. The Build 1 milestone also included morphing-based updates to aerodynamic and structural models. The morphing functionality was implemented through the GMAP application. Lastly, Build 1 included analysis of a High Speed Civil Transport (HSCT) configuration.

Build 2 was comprised of the Geometry, Aerodynamics, Propulsion, Structural, Stability & Control, and Mission Modules integrated in ModelCenter. The Geometry, Aerodynamics, and Structural Modules were enhanced whereas the Stability & Control and Mission Modules were added. RapidFEM is included as a Geometry Module option to perform automatic finite element model (FEM) generation. ModelCenter was updated to include a new DataObject functionality. Lastly, Build 2 utilized all of the feature upgrades and new modules as well as pre-existing functionality to execute an updated analysis of the HSCT configuration within the enhanced ModelCenter/Analysis Server environment.

Build 3 added a Noise Module to the system. Additionally the Aerodynamics Module was enhanced to support multi-fidelity aerodynamics. Build 3.1 added parametric definition of the planform to the Structural Module through use of RapidFEM. The Structural Module is now capable of generating finite element models based on parametric definitions of the structure. Build 4 will be the final build of the HFMDO System. Build 4 will implement a new configuration and address the need to balance a high fidelity MDAO system with robust system design. A full description of the Build 3.1 modules and their features is discussed in Section III.

B. Design Process

The Supersonic Vehicle Design Process is shown in Figure 2. This standard Design Structure Matrix format shows the analysis modules as blue boxes on the diagonal of the matrix, and the data items used by or generated by the modules are shown as yellow boxes. The far left column of yellow boxes represents inputs to the entire process, and the far right column represents outputs from the process. Otherwise, the outputs of a module are shown on the same row as the module, and the inputs are shown on the same column (for example, FEM Mesh is an output of the Geometry Module and an input to the Structural Module).

Module execution is shuffled to get as much information as possible into the upper-right triangle of the matrix, which represents a feed-forward path, where the module generating the data is executed prior to the module using the data. Feedback paths are possible, but require special consideration (e.g., iteration to convergence). Feedback paths require an initial guess at the data to get started, and the estimate is updated as the modules are executed. The ordering of the modules should be tailored such that the number, relative impact, and ease of developing initial starting guesses for the feedback communication paths is minimized. Note that Figure 2 only shows the most significant interactions between modules.

In this process, the Geometry Module takes the geometric variables and generates an updated CFD model (through mesh morphing), a FEM mesh (through parametric geometry & meshing), and information for the Propulsion Module (through simple equations). The Propulsion Module calculates the propulsion performance for use in mission simulation, takeoff/landing analysis, and stability & control. The Aerodynamics Module calculates the vehicle aerodynamic coefficients and distributed pressures at various flight conditions for use in performance simulation and loads calculation. The Structural Module calculates loads and structural sizing to estimate structural weight, which is passed to the Mission Performance Module to estimate the takeoff gross weight (TOGW). The Stability & Control Module determines available control authority for sufficient trim and adequate static stability characteristics. The Mission Module, which currently only calculates Takeoff/Landing parameters, will verify that the vehicle can take off and land on the required fields. In a future build when the Noise Module is incorporated, the resulting takeoff flight paths will be passed to the Airport Noise Module to calculate the noise footprint of the vehicle.

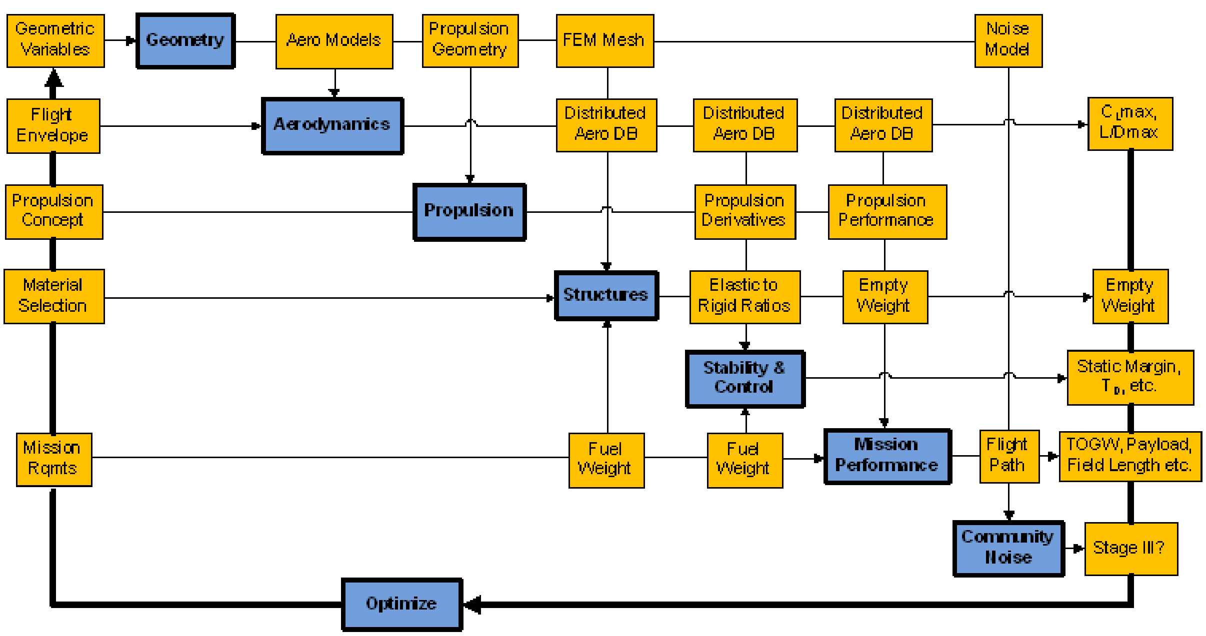

Figure 2. Supersonic Fixed Wing Vehicle Process.

III. HFMDO System

Python modules were integrated into the ModelCenter framework to create the HFMDO System. The framework and the modules are discussed below.

A. HFMDO Framework

In high fidelity analyses, key physical quantities are not well represented by individual variables; items such as loads, boundary conditions, pressure distributions, and displacement fields are made of many (sometimes thousands of) individual variables. Engineers naturally think in terms of these overall physical quantities, and obviously want to be able to manipulate these variables as a unit. To better support needs of high fidelity analyses, new capabilities were added to ModelCenter that allow users to define their own custom data types. Using the custom data types, users can create “wing” or “engine” variables instead of dealing with individual parameters. This object oriented approach simplifies tool wrapping and interdisciplinary coupling. Because individual analyses may be executing on different computer platforms and using different computer languages, these objects must be independent of computer language and platform.

Geometry objects and the aerodynamic database of the HFMDO system were converted to ModelCenter’s custom data types. The HFMDO objects are transferred in the language neutral XML format. The hierarchical data structure of the objects is represented inside ModelCenter, so that users can examine and modify them in a transparent manner. The remainder of this section describes the new Custom Data Type infrastructure and its application.

1. Development of the Data Object Infrastructure

Following the paradigm of object oriented programming, the Custom Data Type Infrastructure makes a distinction between the definitions of data types (classes) and their instances (objects). One benefit of this approach is that the class definition can be reused and shared by many analysis modules. A group of related data items can be linked and consistently transferred using the class information.

A graphical “Class Library” was created to allow users to define custom data types directly from the ModelCenter user interface (Figure 3). A custom data type consists of primitive data members (numbers, booleans, and strings) and arrays of these primitives. Data members can be arranged hierarchically using groups. Meta data such as description, units, and bounds can also be defined for each data member. The Class Library automatically generates Java or Python class files under the hood based on user input. Because they are regular script files, users can modify the generated class file to customize its behavior. For example, utility methods can be added to perform simple computations (e.g., compute wing aspect ratio).

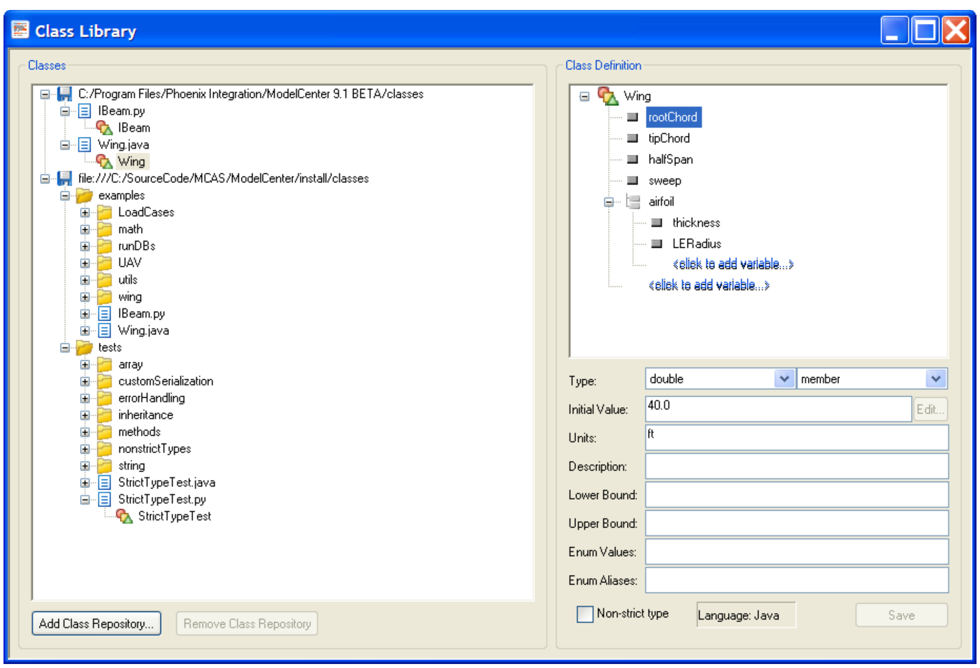

Figure 3. Use Class Library to browse and define a custom data type.

To create an object variable in ModelCenter, users can browse available class locations and select one. The hierarchical structure of the object can be examined in the variable tree, and its member values can be edited (Figure 4). ModelCenter's Link Editor was updated to create links between objects and other variables. One object can be linked to another to pass the entire data structure in one link. Individual data members can also be linked to other data members or to other primitive variables. Because a link can be created between two objects, the use of data objects can dramatically reduce the number of links that need to be created and maintained. This reduces both maintenance costs and user errors when complex data needs to be transferred between components.

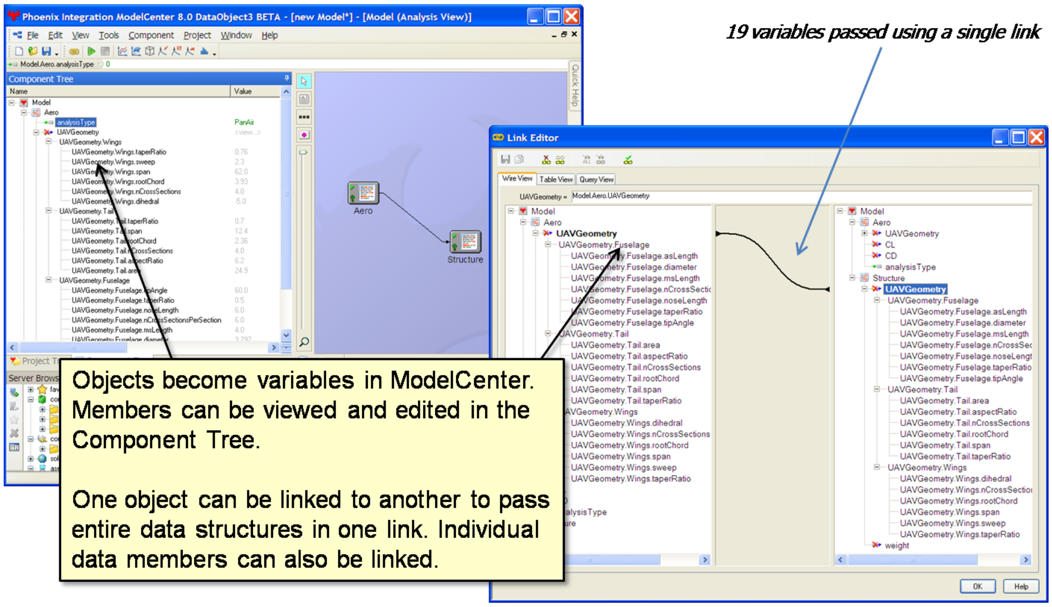

Figure 4. Custom data objects are accessible from Component Tree and links can be created using Link Editor.

2. Development of HFMDO Objects Using the Custom Data Type Infrastructure

The Custom Data Type capability was used to model HFMDO objects such as Geometry Calculations Object, GMAP morphing object, and aerodynamic database object. Custom XML serialization was used for each object to define public data members that will show up in ModelCenter and private data members that are internal to the object. Figure 5 shows the aerodynamic database object created in ModelCenter. The internal data structure of the object is exposed in the variable tree allowing both input and output variable values to be viewed and input variable values to be optionally modified. The aerodynamic database object was transferred from the aerodynamic component to the range calculation component.

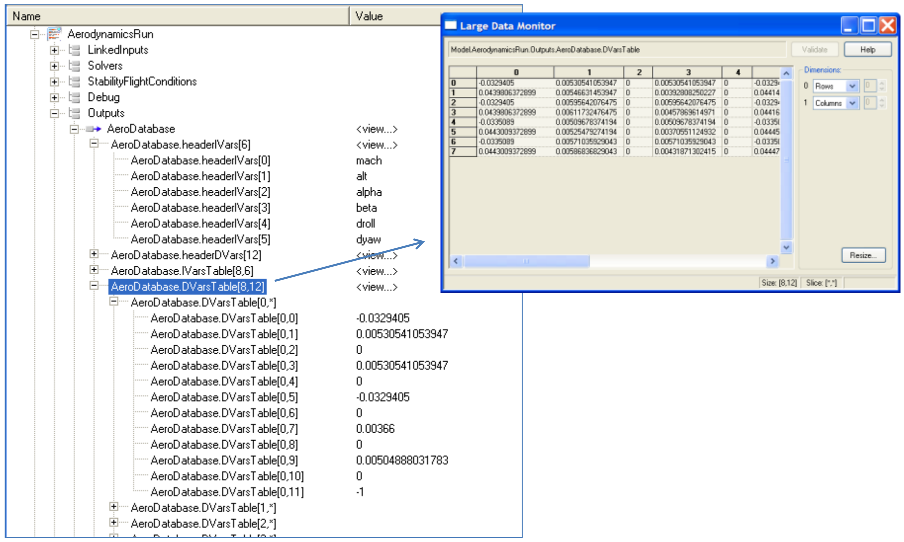

Figure 5. Within ModelCenter, users can browse content of the aerodynamic database object, which is transferred to other components over the network.

B. HFMDO Modules

1. Geometry Module

The Geometry Module generates a finite element mesh based on an OML and a sketch file, modifies a user-supplied mesh by moving its grid points based on changes in supplied design variables, and calculates various geometric properties of the mesh. The Geometry Module takes in two basic types of inputs: configuration information (where input models are located), and geometric design variables (e.g., Sweep, Wing Area, Aspect Ratio, etc.). It generates two objects: a Geometry Object and a Geometry Calculations Object. These two objects are passed to other modules in the system, so that those downstream modules can call methods on the objects to morph geometry (Geometry Object), generate a FEM (Geometry Object), or calculate geometric values (Geometry Calculations Object), such as the length of the vehicle, the area of the wing, and the Aspect Ratio. This allows models to be easily generated as necessary, rather than requiring that all information is defined at the beginning of the execution process.

2. Aerodynamics Module

The Aerodynamics Module calculates aerodynamics (in the form of an Aerodynamic Database Object) for a vehicle using the low fidelity, potential flow code, Panair, the high fidelity, inviscid, unstructured code, Cart3D, and the high fidelity, viscous or inviscid, unstructured code, Usm3D. This module uses the Geometry Object to call GMAP to morph a baseline mesh as design variables change. Additionally, the Aerodynamics Module calculates C_{Lmax} for the vehicle as well as the stability derivatives of the vehicle (in the form of a Stability Derivative Database Object) using the empirical code, Datcom.

The Aero Module is capable of running multiple aerodynamic analysis codes: Panair, Cart3D, and Usm3D may be executed. The Aerodynamics Module accomplishes its analyses by first building a structured list of flight conditions to run. Each analysis case must be placed in a “box”, which defines which code is going to run the analysis point. A request to the Geometry Object is made to morph the grid before the aerodynamic code (Panair, Cart3D, or Usm3D) is run. The aerodynamic program code (such as the Panair code) is responsible for taking a list of models and flight conditions to run. The results are returned and the next analysis code may be run.

A new feature of the Aerodynamics Module is the use of multi-fidelity aerodynamics to create a mid fidelity result. Traditionally, an Aero Database is constructed for use by downstream modules to calculate performance. The user can run many low fidelity results, which execute quickly, capture the trend, but have low accuracy. Or the user can fun a few high fidelity results which take longer to execute, do not capture the trend, but have high accuracy. The mid fidelity approach allows the user to run a few high fidelity results and many low fidelity results. An error database is constructed by interpolating over the low fidelity results and correcting them with the high fidelity results. A mid fidelity database is then constructed by correcting the low fidelity database with the error database. It is this mid fidelity database that is then used by downstream modules. It can be interpolated over to find C_L, C_D. C_{MY}, etc. Time is saved, the trend is captured and accuracy is improved.

The resulting Aero Database and Stability Derivative Database are objects, which have the capability of interpolating on the results and querying any previously analyzed cases. The resulting pressure databases are not objects, but may be viewed by a standard viewer. Although the pressure databases are not objects, the Aero Database Object has methods that operate on the pressure databases and can perform operations on those files. An example of an operation that is supported is interpolation.

The Cart3D HSCT aerodynamic model was created by taking the Panair/Plot3D formatted mesh and triangulating each quadrilateral. Then, the mesh was mirrored and duplicate triangles were removed as Cart3D requires a watertight mesh with no duplicate elements. The Cart3D aerodynamic mesh is shown below in Figure 6.

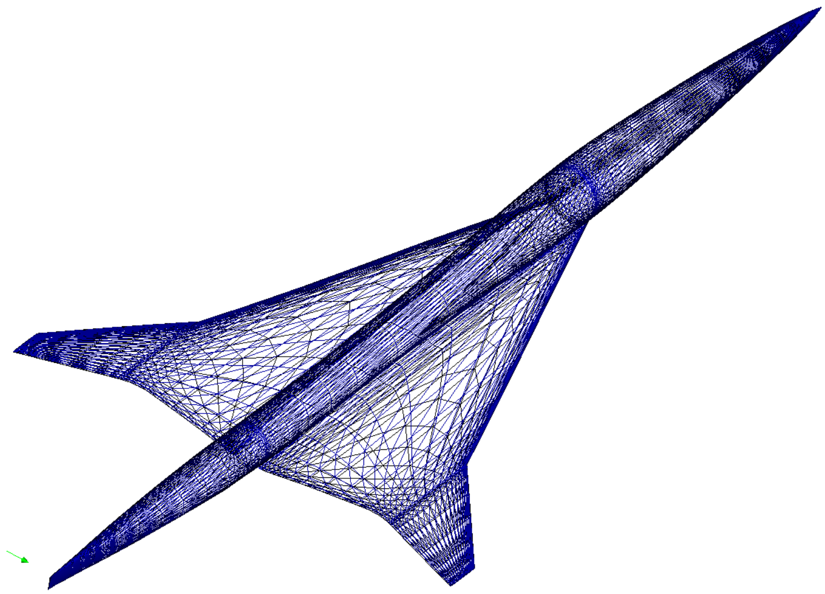

Figure 6. Cart3D Aerodynamic Mesh.

3. *Propulsion Module*

The Propulsion Module calculates the overall performance of an engine under a given flight condition. The Propulsion Module is based on NPSS and utilizes an NPSS engine model of a small business jet. This engine model has the same cycle as the engine used in the High Speed Civil Transport (HSCT) and was scaled up to the HSCT engine thrust level.

4. *Structural Module*

The Structural Module uses MSC/NASTRAN to analyze either a pre-made structural bulk data file (bdf) or a bdf assembled using M4 Engineering's RapidFEM software. The module can use a fully stressed design (FSD) technique or NASTRAN's gradient optimization technique to determine the weight optimized structural sizing for the model. Lastly, the module determines mass properties of the structure and property regions as well as provides a margin summary of the optimized structure. As design variables change (i.e., a trade study execution) the Structural Module can invoke RapidFEM by utilizing the Geometry Object to generate and then morph a new finite element model; or the Structural Module can use the Geometry Object to call GMAP[5] to morph an existing finite element model. The following combinations of model input and optimization techniques have been executed: 1) User supplied FEM running Fully Stressed Design (FSD) optimization, 2) User supplied FEM running gradient based optimization (NASTRAN Solution 200) and 3) RapidFEM generated FEM running gradient based optimization.

For Build 1.2, the Structural Module executed the RapidFEM capability within the Geometry Object as the pre-processor to create the finite element model with the linear aero model and analysis conditions. Build 2 added improved body meshing and FEM merge/trim capabilities. Figure 7 shows the model generated by RapidFEM. The outer mold lines of the S4T wind tunnel test model components were scaled to full size and used as inputs to RapidFEM. The internal structure was defined as a reasonable layout of ribs and spars in the wing/tail and floor and bulkheads in the fuselage. The fuel and payload were added based on the mass properties described in Reference 8.

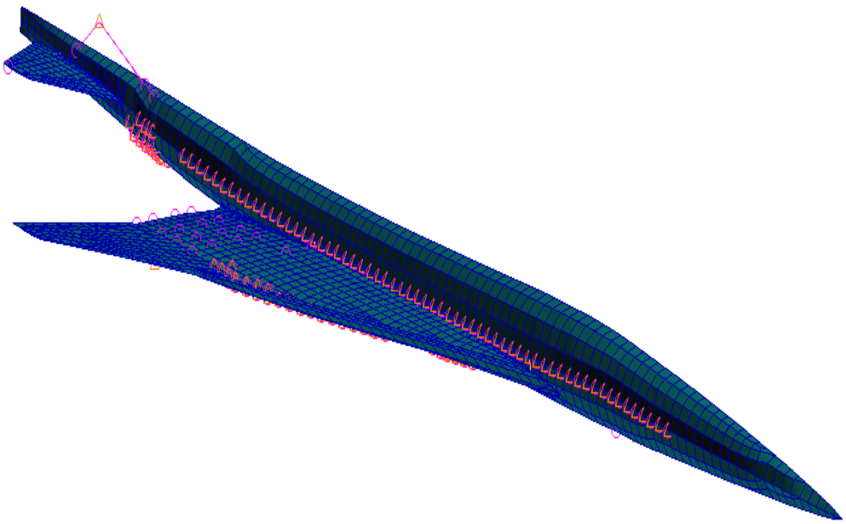

Figure 7. RapidFEM Generated HSCT Structural Model.

Build 3.1 demonstrated a significant step forward in making models of airframe structures with the Structural Module utilizing the enhanced RapidFEM tool. Previously, RapidFEM required the analyst to explicitly define the planform. In Build 3.1 using RapidFEM's new smart structure, the entire planform can be described parametrically and altered with minimal input. Components can be divided into sections by type of layout. Structural properties can also be defined based on a map of the structure. The wing structure options include spar orientation, rib orientation, and distribution. The spar can be parallel to the leading edge, parallel to the trailing edge, or root and tip of spar at constant percent chord. The rib orientation can be perpendicular to the leading edge, perpendicular to the trailing edge, or constant span. The distribution can be fixed number, fixed spacing or a specified distribution. The fuselage structure options include the floor, bulkhead, and frame. Only one floor can be defined in a section with smart inputs and the placement is either flat (constant Z) or constant percent height. The bulkhead is located at specific fractions of the section length and they are flat (constant X). The frames can be ring or rail. The frame distribution can be fixed number, fixed spacing, or a specific distribution. Figure 8 shows how a wing with a complicated planform can be divided into sections. Figure 9 shows a FEM of the HSCT wing generated using RapidFEM's parametric definition of the planform.

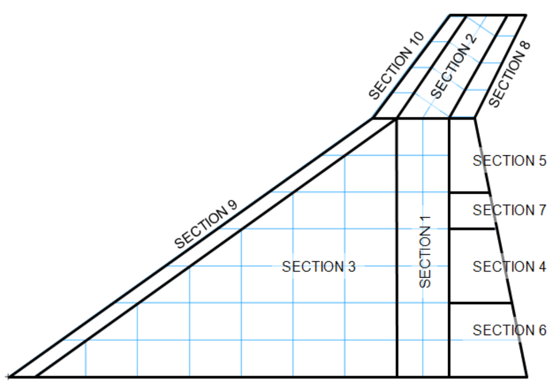

Figure 8. Complicated Planform Divided into Sections

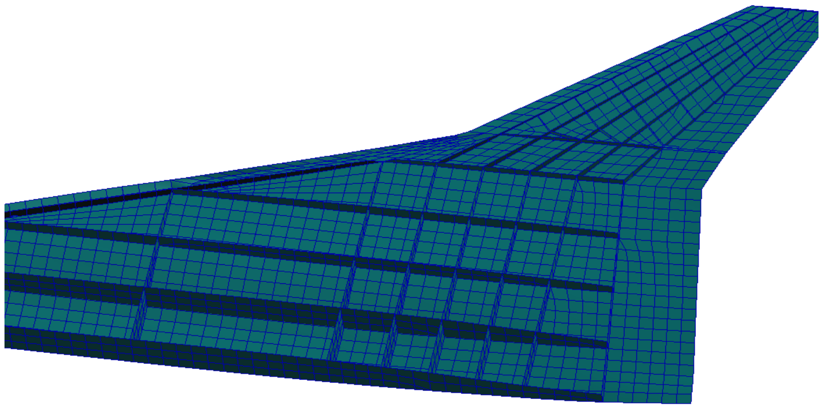

Figure 9. RapidFEM Generated FEM showing Wing Internal Structure

5. Stability and Control Module

The Stability & Control Module is primarily responsible for the analysis and design of the stability and control characteristics of a user-supplied aircraft model for a number of critical flight conditions. The module uses the Matlab Stability and Control Toolbox (MASCOT) to perform a low fidelity stability and control assessment of the aircraft model. The module determines available control authority for sufficient trim and adequate static stability characteristics.

In Build 2, the Flight Critical Condition (FCC) Set No. 1 horizontal flight condition is tested. The S&C Module utilizes the Aerodynamics Module's DATCOM database to calculate the aerodynamic coefficient and derivative data required for the S&C analysis. As DATCOM solves only subsonic problems, the Set No. 1 flight conditions are set to Mach = 0.8, alt = 35,000 feet, and attack angle $\alpha = 0^0$. The Set No.1 S&C analysis is treated as a three degree of freedom (3 DOF) problem. Six degree of freedom (6 DOF) lateral analyses are not performed in Build 2.0 and may be a future enhancement to the module.

6. Mission Module

The Mission Module is responsible for calculating the takeoff and landing performance characteristics of a given aircraft configuration. Note that this implementation of the Mission Module uses TAKEOFF, which is a stand-alone version of Flight Optimization System (FLOPS). TAKEOFF only analyzes takeoff and landing performance. It has been validated that the Mission Module successfully wraps TAKEOFF and outputs the results in a convenient form for the user without degrading any accuracy from the initial TAKEOFF stand-alone program results.

7. Noise Module

The Noise Module is responsible for calculating the takeoff and landing performance characteristics of a given aircraft configuration. The Noise Module is implemented using Aircraft Noise Prediction Program (ANOPP). Specifically, the module calculates the effective perceived noise levels and predicts whether or not the aircraft meets the standards specified by FAR 36. The effective perceived noise levels include sideline, approach, and takeoff flyover. The output is a Pass/Fail flag for FAR 36 Stage III Certification.

IV. Example Problems and Results

A. System Setup

Each of the modules within the HFMDO System was developed in Python. ModelCenter was used in order to integrate the modules into a system framework. To do this, scriptWrapper files were developed to link the Python code from the modules into ModelCenter components. The system is shown in Figure 10.

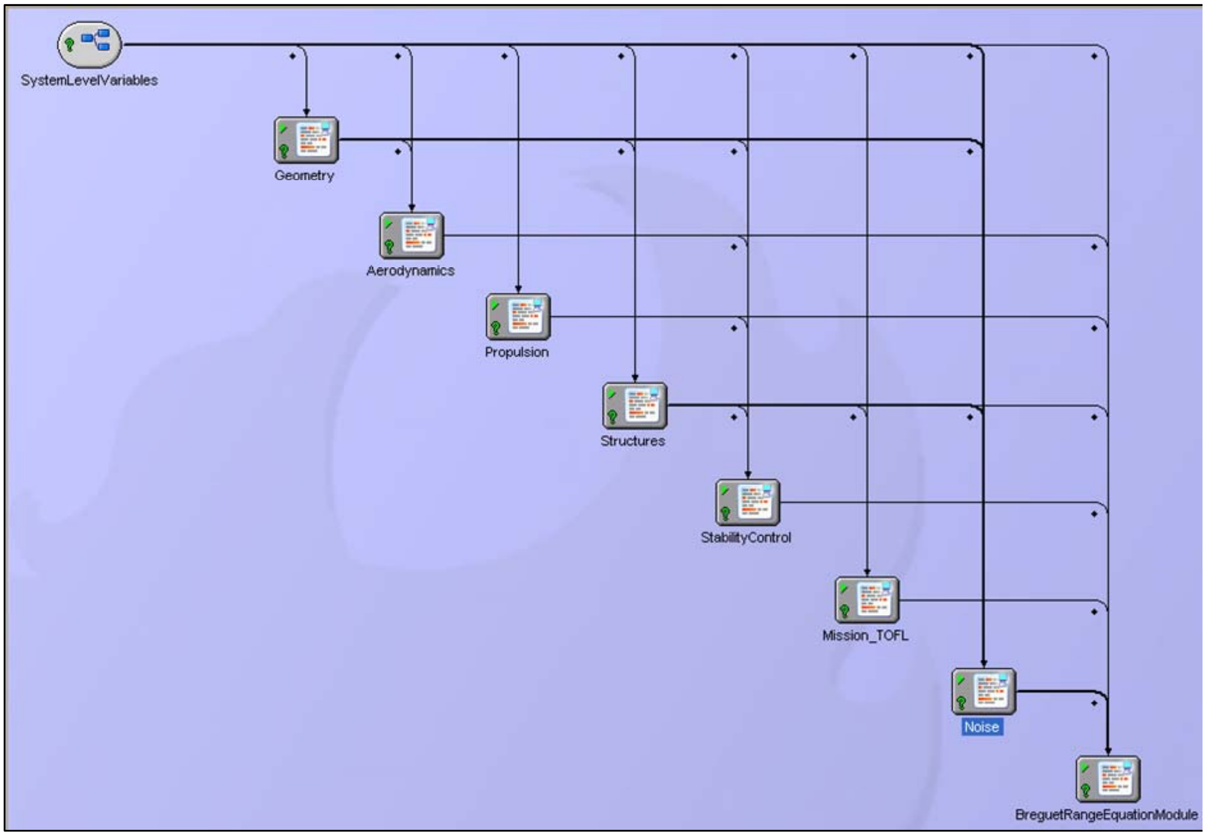

Figure 10. HFMDO System Integrated in ModelCenter.

The eight main components in the system model are:

1. SystemLevelVariables
2. Geometry
3. Aerodynamics
4. Propulsion
5. Structures
6. StabilityControl
7. Mission_TOFL
8. Noise
9. BreguetRangeEquationModule

The Breguet range equation shown in Eq. (1) was used to calculate the range from a variety of system level outputs:

- Range – Range computed from Breguet range equation
- V – Velocity at current Mach number and altitude
- L/D – Lift/Drag
- Isp – Specific Impulse
- W – Weight

$$Range = V \frac{L}{D} I_{sp} ln\left(\frac{W_{initial}}{W_{final}}\right) \quad (1)$$

B. Example Problem – HSCT Configuration

The ModelCenter framework is responsible for coordinating the execution of the HFMDO System. This includes ensuring that all the required modules are executed in the correct order, that the required data are transferred from one module to the next, and that the overall trade study or optimization process is carried out efficiently. Build 2 uses two types of execution options: Single Point Analysis and Parametric Trade Study. Single Point Analysis is a single run through the HFMDO system to evaluate the performance of the current design. Parametric Trade Study allows the user to define ranges for the system design variables; and, the HFMDO system loops over the defined ranges, evaluating each design point. This allows the sensitivity of the design to variations in continuous-valued design variables to be quickly evaluated.

For this HSCT example problem, three trade studies were examined in which a geometric design variable was varied. The three geometric design variables studied were Sweep, Wing Area, and Aspect Ratio. These design variables were plotted against Range, L/D, and TOGW. Table 1 summarizes the data collected from the studies. Note that the red boxes represent a design point in which the NASTRAN optimization did not converge on a final value.

Table 1. Trade Study Results

	Sweep Study					Wing Area Study					Aspect Ratio Study				
Parameter	Run 1 Values	Run 2 Values	Run 3 Values	Run 4 Values	Run 5 Values	Run 1 Values	Run 2 Values	Run 3 Values	Run 4 Values	Run 5 Values	Run 1 Values	Run 2 Values	Run 3 Values	Run 4 Values	Run 5 Value
Wing Area (in^2)	1235285	1235285	1235285	1235285	1235285	1000000	1200000	1400000	1600000	1800000	1235285	1235285	1235285	1235285	123528
AR	2.008	2.008	2.008	2.008	2.008	2.008	2.008	2.008	2.008	2.008	1.500	1.775	2.050	2.325	2.600
Sweep (°)	64	66.5	69	71.5	74	66.1	66.1	66.1	66.1	66.1	66.1	66.1	66.1	66.1	66.1
Taper	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074
Omega	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610	0.610
Eta	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619	0.619
Theta	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765
Range (nm)	4426.285	4370.087	4469.28	4608.483	5136.197	4582.913	4381.694	4217.641	3961.499	3784.423	4536.131	4372.226	4320.612	4295.128	4092.7
L/D	11.11151	11.23811	11.42715	11.80713	12.54938	11.20378	11.22286	11.21739	11.19442	11.03413	11.46009	11.33027	11.2066	11.07546	10.941
Isp (s)	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.45	3618.4
TOGW (lb)	802952.3	814497.2	811711.6	812695.8	790324	790581.1	812557	831097.1	862446.1	879408.9	805978.2	818234	818673.9	815819.1	833668
1/2 TOGW (lb)	401476.1	407248.6	405855.8	406347.9	395162	395290.6	406278.5	415548.5	431223.1	439704.5	402989.1	409117	409337	407909.6	416834

C. Optimization Problem

By utilizing the HFMDO system, an optimization problem to maximize range was posed. This problem sought to maximize range subject to constraints on vehicle weight and takeoff field length. The design space for this problem was created by varying wing sweep and aspect ratio. A summary of the constraints and final optimized design values are shown in Table 2. Figure 11 shows a comparison of the baseline and optimized configuration vehicle models. It is important note that there remains room for improvement in the quality of the result, which could be obtained by modifying the NASTRAN optimization parameters.

Table 2. Summary of Optimization Results

AR	Sweep (deg)	Range (nm)	TOFL (ft)	Weight (lb)	Weight Constraint (lb)	TOFL Constraint (ft)
1.777	71.500	4662.5	12500.0	809376	821800	12500

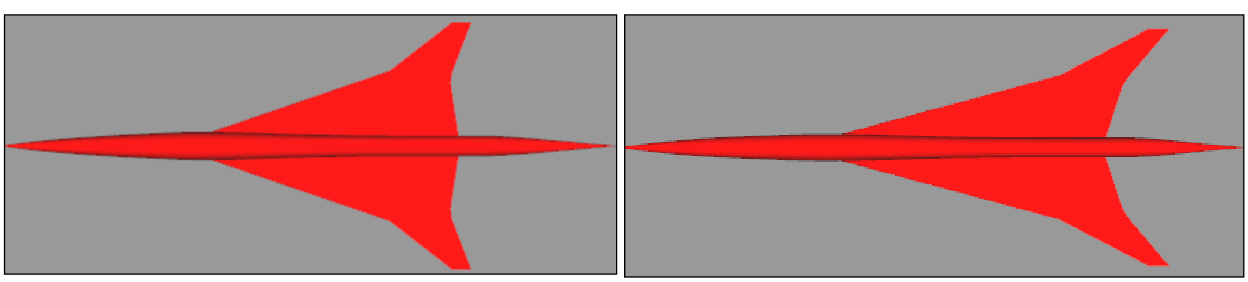

Figure 11. Left: Baseline Model, Right: Optimized Model.

D. Mid Fidelity Aerodynamics Example

An example problem was executed using the HSCT configuration to demonstrate the mid fidelity aerodynamics capability. 60 low fidelity (Panair) and 18 high fidelity (Cart3D) cases were run. This example covers the subsonic, supersonic flight regimes. The Panair mesh typically took no longer than 2 minutes per run. The small Cart3D mesh took about 15 minutes per run while using 8 processors. Note that if a dense Cart3D mesh was used, it would have taken about 60 minutes per run. The mid fidelity execution took 5.1 hours while the high fidelity execution took 14.6 hours for 60 cases run. The mid fidelity option resulted in a 65% reduction in runtime. In Figure 12, the results for $C_{L\alpha}$ vs Mach are shown. The trend is reasonable for high fidelity and low fidelity runs. Figure 12 also shows the results for C_{Do} versus Mach at 55K feet. The high fidelity results "pave over" the transonic and supersonic flight regimes and the trend is not captured. This is corrected by the low fidelity results, which were not as accurate but captured the trend well. This example shows the usefulness of the mid fidelity approach to obtain reasonably good results with a compromise between time and accuracy.

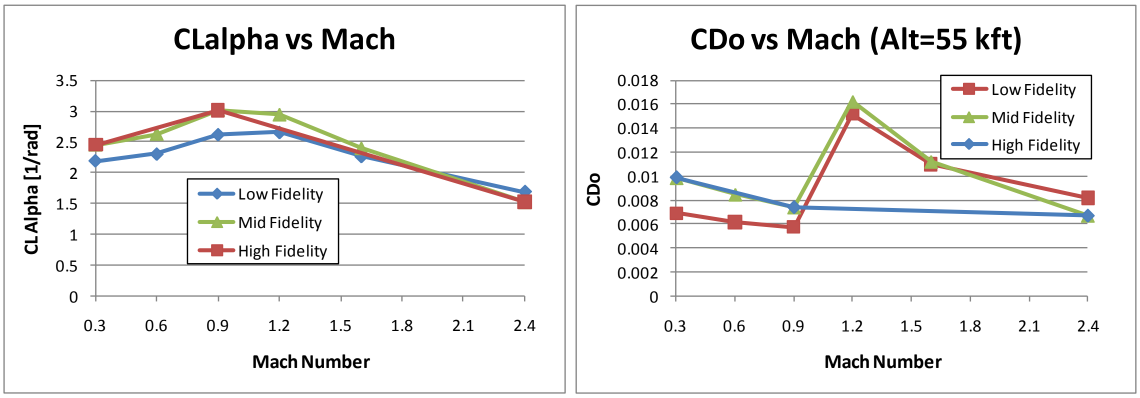

Figure 12. Mid Fidelity Results using Multi-fidelity Aerodynamics

E. RapidFEM Example

An example problem was executed to demonstrate the improved RapidFEM functionality of creating a complete static aeroelastic with gradient based optimization analysis using parametrically defined meta-mesh inputs. Using the scaled S4T OML and RapidFEM parametric sketch files, a half model was built to analyze a 2.5 g pull-up flight event. Composite properties were used for the wing, horizontal tail and fuselage structure. The ply thicknesses of the wing were optimized using NASTRAN's gradient based optimization technique (SOL 200). The fuel and payload were included by applying concentrated masses to the tanks and floor. The engines, landing gear and vertical tail were modeled with concentrated masses. The Structural Module used this FEM to determine the optimized vehicle weight. The displacement and strain results for the initial and optimized models are shown in Figure 13.

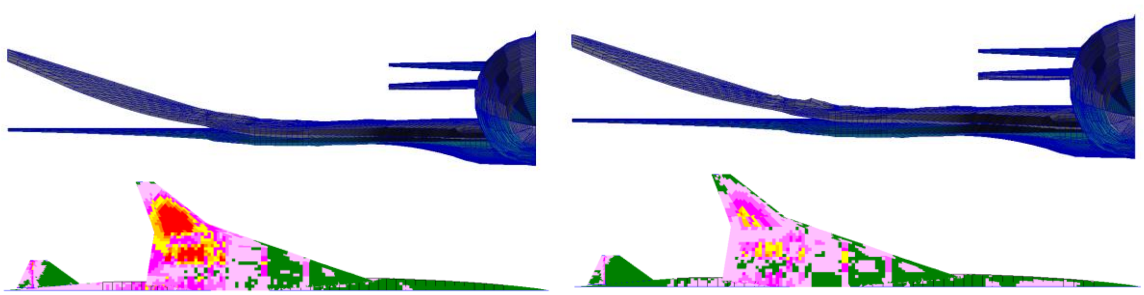

Figure 13. Structural Module Initial and Optimized Results

V. Conclusion

The development of the HFMDO analysis tool is nearing completion. The system has been integrated into the ModelCenter framework. Capabilities have been implemented to define, create, and transfer custom data objects using ModelCenter and Analysis Server. Additional capabilities such as multi-fidelity aerodynamics and rapidly generating FEMS with smart structure are recent enhancements. These recent enhancements showcase high fidelity features that are not readily available in today's MDAO systems. The HFMDO has evolved into a high fidelity analysis tool suite suitable for conceptual and preliminary design.

As HFMDO uses an incremental build approach, there is one last build to be completed. Build 4 is the final build and it will be complete in December 2010. Future work[9] will include enhancements to ModelCenter and Analysis Server for improved support of data object and file data type. The Custom Data Type infrastructure will be updated to support file and file array members so that files can be transferred as an attribute of an object. To support creation and maintenance of complex models, capability to store and share commonly used class definitions using a server will be added. Future work will also include switching configurations and focusing on robust system design while retaining high fidelity functionality. High fidelity analyses are inherently tightly coupled. It is important to be able to easily change configurations and execute them without extensive rework to the system. This goal will be achieved by working towards system and configuration separation. High fidelity functionality will be balanced against a robust system design that allows for enough flexibility such that interdisciplinary couplings do not cause an inability to easily modify and change configurations.

Acknowledgments

Funding for the development of HFMDO is provided by NASA Langley Research Center, Fundamental Aeronautics Program, Supersonics Project, Contract # NNL08AA00C. The authors also wish to acknowledge the rest of the HFMDO team for their contributions to this work.

References

[1]Baker, M. L., Munson, M. J., and Alston, K. Y. "*Integrated Hypersonic Aeromechanics Tool (IHAT),*" AIAA Missile Sciences Conference, Monterey, CA, November 2004.

[2]Phoenix Integration, Inc. *"Improving the Engineering Process with Software Integration",* A Technical White Paper, 2002.

[3]M4 Engineering, Inc, *Multidisciplinary Optimization Object Library (MOOL) - Module Reference Manual 2.0,* December, 2009.

[4] *MSC. Nastran 2005 Quick Reference Guide: Volume 1*, MSC Software Corporation, Santa Ana, CA, 2004.

[5]Raymondson, C. R., et al. *"Geometry Manipulation by Automatic Parameterization (GMAP)".* AIAA MDAO Conference, September 2008.

[6]Dahlin, A. N., Baker, M. L. *"GMAP Mesh User's Manual"*, M4 Engineering, June 2010.

[7]Stuewe, D. T., Baker, M. L. *" RapidFEM User's Manual"*, M4 Engineering, June 2010.

[8] *High Speed Research Program HSR II – Airframe Task 20, Task 2.1 – Technology Integration, Sub-task 2.1.1.2 December 1997 PTC*, February 1998.

[9]*Fundamental Aeronautics Program Multidisciplinary Analysis and Optimization Software Requirements Specification*, MDAO-SRS, January 31, 2008.