AIAA SciTech Forum
13-17 January 2014, National Harbor, Maryland
52nd Aerospace Sciences Meeting

10.2514/6.2014-0188

Computational Geometry for Multifidelity and Multidisciplinary Analysis and Optimization

Nitin Bhagat[1]
Universal Technology Corporation, Fairborn, OH 45432

Edward Alyanak[2]
Wright Patterson Air Force Base, AFRL, WPAFB, OH 45433

The multidisciplinary analysis and design optimization process requires an automated work flow that must include the seamless integration of geometry, meshing, and analysis tools. The goal of this effort is to create an automatic geometry model generation tool that supports automatic meshing for analysis tools, and to satisfy the demands of multifidelity, multidisciplinary analysis. The current status of this developing capability and subsequent steps for future improvement are reported. The results of this work are demonstrated with an example that explores the multifidelity analysis of tailless yaw control devices. This example was chosen to stretch this tool by including complex control surfaces, inlets, engines, tails, wings, a canard, and a fuselage. The geometry tool and its capability were successfully demonstrated by using this nontrivial example test case.

I. Introduction

RECENT work within the Multidisciplinary Science & Technology Center (MSTC) at the U.S. Air Force Research Laboratory has been to explore technologies that enable rapid modeling for design in support of automated multidisciplinary design optimization. This effort uses CAD-free methods that support linear, or transpiration flat-plate-based, aerodynamic analyses as well as full Finite Element Modeling.[1] MSTC is currently working to develop capabilities that can support aircraft design that utilizes high-fidelity physics simulations. Due to the high-fidelity nature of the analyses involved, this future capability will be CAD based. The current status of the MSTC investment in this technology, and its use is presented in this paper.

The idea of using realistic geometry for analysis and optimization has been extensively discussed and reported in the literature.[2-4] Previously, limited computational power and appropriate available software were the main hindrances in solving these problems. With the recent hardware improvements and software developments, it now seems possible to perform realistic analysis on a complex geometry. However, a gap still exists between creating geometry and analyzing geometry. This gap is due to the process first creating the geometry, then transferring the geometry information from a CAD-based geometry to a meshing tool, and then finally to the appropriate analysis tools. Different (sometimes proprietary) formats make this three-step transfer process more difficult for seamless interfacing/integration.

With closing this gap in mind, an open-source CAD modeling path is being explored. The Engineering Sketch Pad (ESP) software, jointly developed by Haimes and Dannenhoffer[5] is used to generate CAD geometry models. ESP is dependent on OpenCASCADE[6] for the geometry kernel. Among other things, OpenCASCADE is an object-oriented, three-dimensional solid modeling tool. One issue with OpenCASCADE is its complex structure. However, ESP hides this complexity by providing a simple 'as English as possible' scripting language (or through APIs) that any user can use to develop a three-dimensional geometry model. Another feature of ESP is that it provides both the top-down and bottom-up approaches to geometry generation. Both approaches are useful in their respective ways depending on the project. The geometry model can be attributed at any level for use by a user or analysis codes. For instance, these attributes can be used to identify individual components, which is necessary to perform the meshing and analysis steps.

[1] Research Engineer, 1270 North Fairfield Rd, Fairborn OH 45432, AIAA member.
[2] Project Engineer, AFRL/RQVC, 2210 8th St. Bldg. 146, Rm. 225, WPAFB, OH 45433, AIAA Young Professional.

To perform aerodynamic analyses using the ESP geometry, the automated meshing of the geometry must be handled. The automated surface meshing is done using the open-source code Bidimensional Anisotropic Mesh Generator[7] (BAMG). The automated volume meshing is done using the open-source code Tetgen.[8] The geometry created by ESP can be exported in the CART3D[9] format for aerodynamic analysis. The details on the implementation and results from this geometry/aerodynamics interface are detailed later in the report. In addition, the capability to export ESP geometry into the open-source Stanford University Unstructured[10] (SU^2) format is currently being developed, and preliminary results of this interface are presented.

II. Aircraft Geometry

This section describes the aircraft geometry model example created using ESP. The aircraft geometry is built in such a way that any component of the aircraft can be taken out or added back based on requirements set by the user or analysis tool. The act of taking a component out is called muting. This means that the geometry model is capable to be used for multifidelity analyses. The current multidisciplinary capability is that the geometry model can be exported for both aerodynamic analysis and/or structural analysis.

The aircraft model is shown in an exploded view in Fig. 1. The geometry was created from a notional 3-view sketch of a fighter. It has the following components: wing, fuselage, canard, tails, inlets, and Spoiler-Slot Deflectors (SSD) for control surfaces. Geometric global parameters in the model can be used to mute and unmute certain components when desired. For example, to perform a simplistic engineering analysis, inlets, spoilers or even tails may not be necessary. These components can be muted by changing the corresponding variable (one per component). In addition, each component is parameterized so that is automatically created based solely on its user-defined parameters. When these parameters are modified the geometry regenerates as required.

The aircraft component parameters can be either geometry-based or design-variable-based. To understand the difference, two examples are presented. For the geometry-based parameters, consider two-dimensional sketches at various stations along the span. These parameters define the geometry component as they are lofted to form a fuselage or perhaps an inlet. For the design-based parameters, consider the wing definition. It can be defined by a certain combination of the chord length(s), the angle of attack, the sweep, the aspect ratio, and the dihedral angle. These parameters are normally thought of as design variables in the aircraft design community. They are equally applicable to a tail or canard geometry.

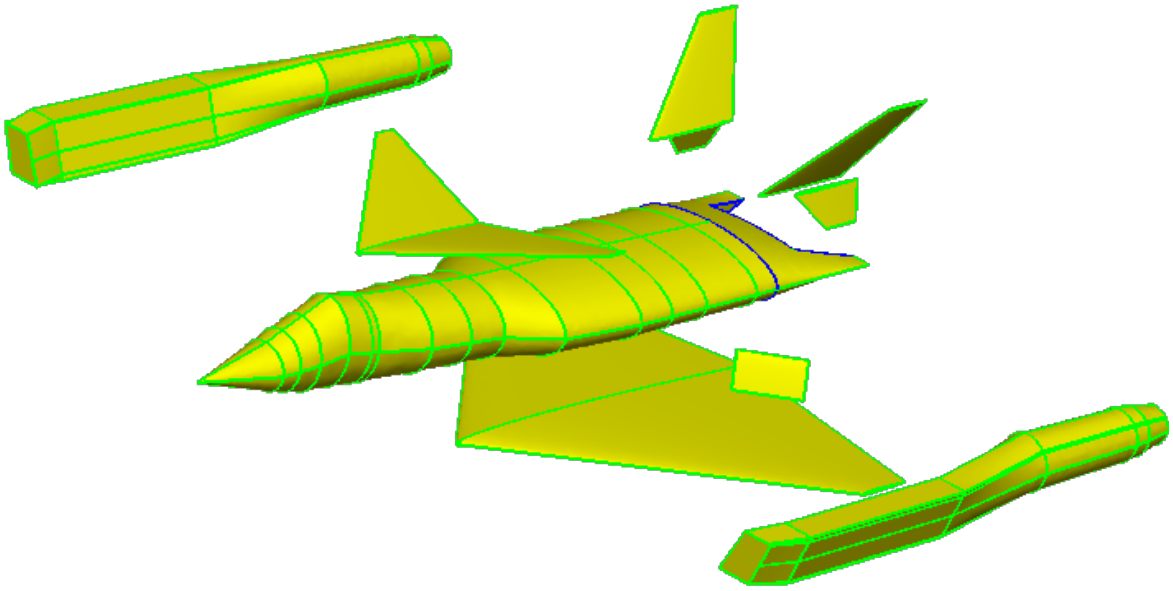


Figure 1: Aircaft geoemtry, component view

The creation of aircraft geometry from parameterized components can be done using the ESP scripting language. For instance, the example geometry from Fig. 1 is built up by ESP with the following simple parameters, available in the script:

```
despmtr   makeWing      1
despmtr   makeSpoilers  1
despmtr   makeFuselage  1
despmtr   makeCanard    1
despmtr   makeTails     1
despmtr   makeInlets    1
```

The key-word, despmtr, behaves like a switch and can have value 0 or 1. If the value is set to 0, that particular component will be muted. The fuselage is built by lofting elliptical cross-sections at appropriate stations along its length. The airfoils for the wing, canard, and tails can be generated by constructing two-dimensional sketches using data/control points provided by the user/designer. These sketches can be piecewise linear curves or splines. Another way of generating the airfoil sections is to build user-defined primitives (UDP). UDPs embed standardized empirical formulations in order to compute data points, such as NACA-4 and -5 series airfoils. This is another useful feature provided by ESP. The canard and tails are built in a similar manner as the wing since they are related geometries. In the case of the canard, different angles-of-attack can be applied to the each side of the fuselage resulting in an asymmetric effect. Similarly, each of the tails can have their own angle-of-attack. For this example, the angles-of-attack for the tails are set such that the top tails and bottom tails rotate synchronously. The inlets are built in a similar manner as was done for the fuselage. This is to loft different sketches at the appropriate cross-sections along the length of the fuselage. The SSDs are created in a similar manner to the wing. Their position, orientation and deflection (which can be varied from 0-90 degrees) are user-defined variables. The complete assembly of the aircraft geometry, when all the components are put together using ESP, is shown in Fig. 2.

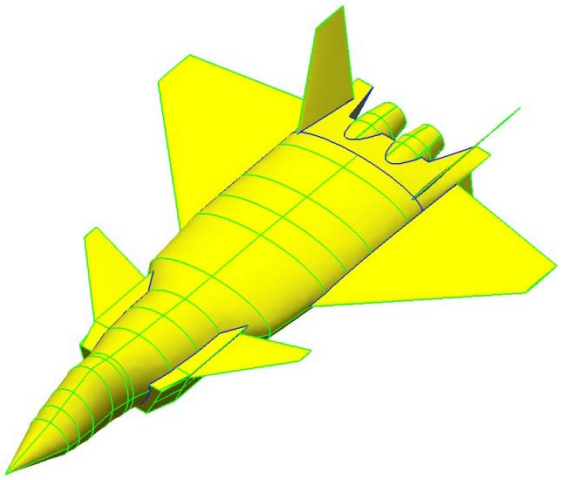
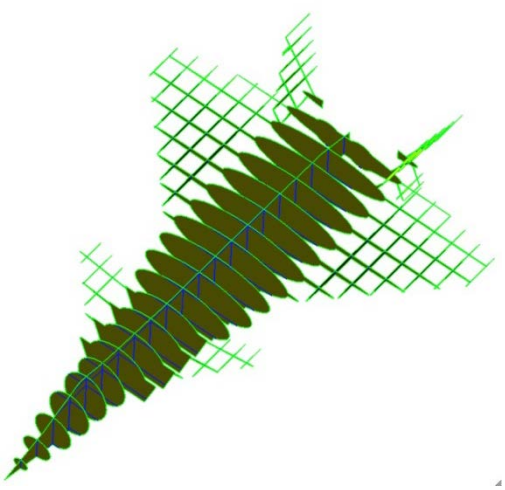


Figure 2a. Geometry: Solid Model, Top View

Figure 2b. Geometry: Solid Model, Bottom View

The assembled solid model shown in Figure 2 can now be used for an aerodynamic analysis. Figure 3 shows a built up internal structural model that was derived from the same aircraft geometry. This model can be used for structural analysis by applying an automatic finite element meshing procedure. This demonstrates the multidisciplinary nature of this CAD-based geometry model.



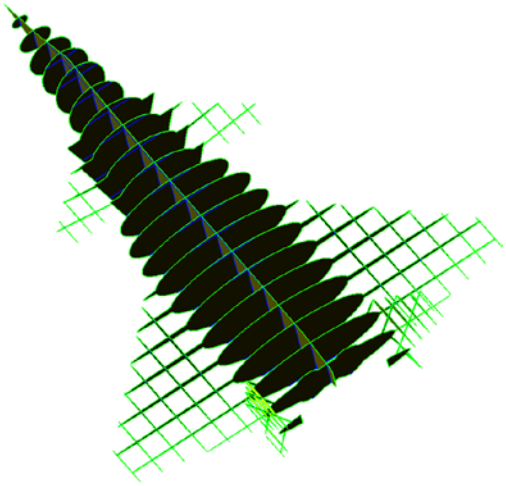
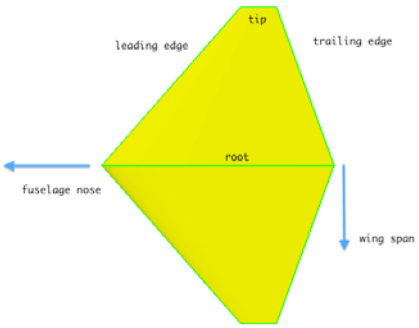


Figure 3a. Internal Structural Model: Top View **Figure 3b. Internal Structural Model: Tail View**

As mentioned before, the aircraft geometry is parameterized in terms of geometry and design variables. This facilitates the ability of this geometry model to reflect any changes in the design variables directly on the geometry. As an example, the parameterization of the wing is illustrated in Fig. 4. The wing can be created in two ways. One way is to simply specify the type of root and tip airfoil, respective chord lengths, their orientation and location of the leading edges. Then, a simple loft between the root and tip airfoil will create the wing geometry. Another way is based on several variables that are used as design parameters to define the wing. These variables are: type of root and tip airfoil, location of the leading edge of the root airfoil, wing area, aspect ratio, taper ratio, sweep location and sweep angle, twist location and twist angle for root and/or tip, and wing dihedral angle. It should be noted that specifying some of these variables might make other variables redundant, so it is assumed that designer/user is aware of the conflicts; only the variables necessary to defining the wing are specified. The dependent variables such as wing span, root and tip chord, and their location is computed using standard formulations.

The orientation of the wing is illustrated in Fig. 4(a). The location of tip airfoil with respect to root airfoil is determined based on the sweep angle (specified in degrees) and sweep location (chord location, line joining root airfoil with tip airfoil, value ranging from 0-1.0). When the value of sweep location is set to 0, the sweep angle provides leading-edge sweep, whereas, when the value is 1, the sweep angle provides trailing-edge sweep. For all other values in the range 0 to 1, the sweep angle is with respect to the location of the sweep along the chord length. Figures 4(a-f) illustrate the results of various combination values for sweep angle and location. For the configuration shown in Fig. 4(a), the wing has a leading-edge sweep of 15 degrees. Figure 4(b) shows a wing that is generated by changing the sweep location to the leading-edge, while leaving the other parameters the same. Figure 4(c) has a quarter chord sweep location, Fig. 4(d) has a mid-chord sweep location, and Fig. 4(e) has a trailing-edge sweep location. Figure 4(f) has a sweep angle of 25 degrees at the trailing edge.



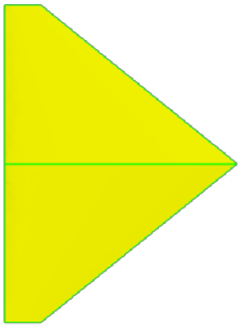


Figure 4a. Sweep angle=-15°, loc=1 **Figure 4b. Sweep angle=0°, loc=0**

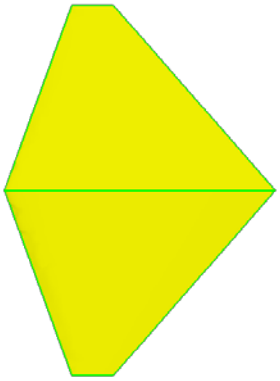


Figure 4.c. Sweep angle=0°, loc=0.25

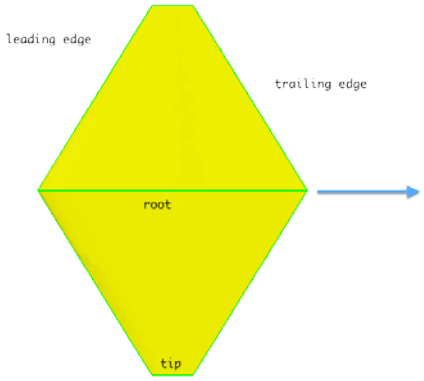


Figure 4d. Sweep angle=0°, loc=0.5

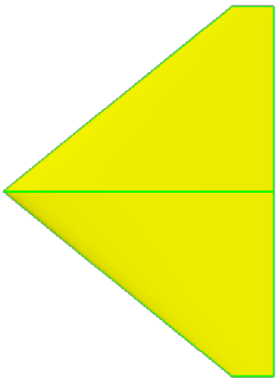


Figure 4e. Sweep angle=0°, loc=1.0

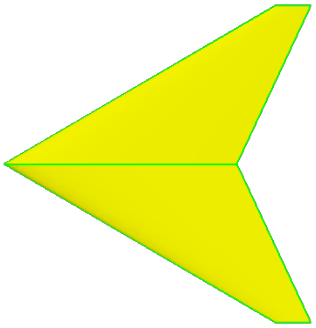


Figure 4f. Sweep angle=25°, loc=1

The wing can have SSDs integrated into it. These control effectors were muted in Fig. 4 for simplistic illustration purposes. The geometry definition of an SSD in the wing is done as follows. The location of an SSD is specified in terms of its relative position from the trailing edge of the tip of the wing, as shown in Fig. 5. The dimensionality of an SSD is determined based on relative percentage to the root wing chord (for length) and half span (for width).

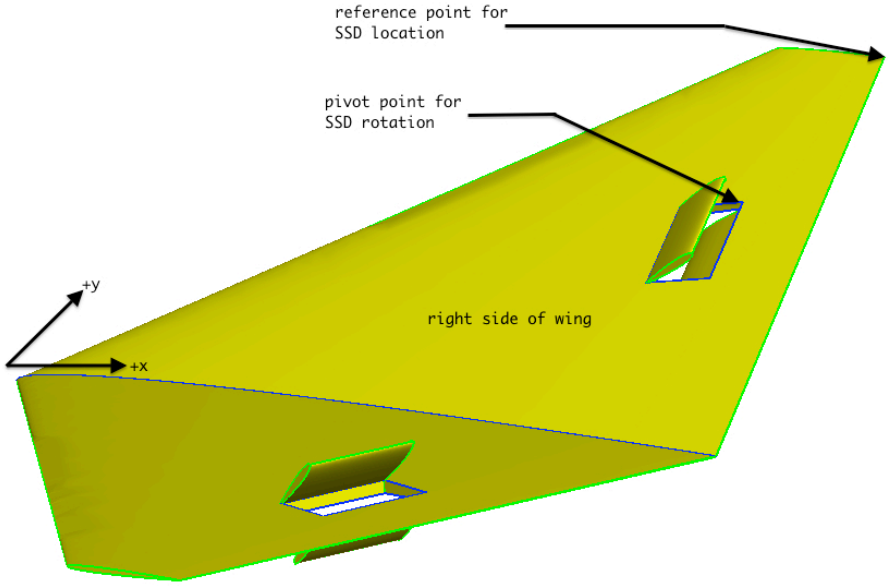


Figure 5: Spoiler-slot deflector, reference location.

SSDs are created in pairs with one being on the left and one being on the right wing. Four locations are required to define its placement on the wing. These locations are the front (towards the leading edge of the wing), the rear (towards the trailing edge of the wing), the top (side of the wing), and the bottom (side of the wing). The deflection angle for the SSDs can be specified independently on the left and the right wing for generality. Figures 6(a-f) illustrate some of the possible configurations. Figure 6(a) shows the wing/SSD configuration where the top-rear and bottom-front spoilers are specified with the deflection angles of 0° and 45° on the right and left, respectively. Figure 6(b) shows the wing/SSD configuration where the top-front spoiler is specified with a deflection angle of 90° on the right side and 45° on the left side. Figure 6(c) shows the wing-SSD configuration where the top-front spoiler is specified with the deflection angle of 15° on the right and on the 90° on the left. Figure 6(d) shows a wing/SSD configuration similar to Fig. 6(c), but includes a 25° wing twist at mid chord. Figure 6(e) shows the wing/SSD configuration where the spoiler deflection angle is set to 15° and angle of orientation about pivot point for the spoiler (as was shown in Fig. 5) is set to -15°, whereas Fig. 6(f) shows the wing/SSD configuration with the same deflection and orientation as in Fig. 6(e). However, the location, length, and width of the spoilers in Fig. 6(e) have been altered. Figures 6(a) through 6(d) have their dihedral angle set to 25 degrees, whereas Fig. 6(e) and Fig. 6(f) have 0° as their dihedral angle.

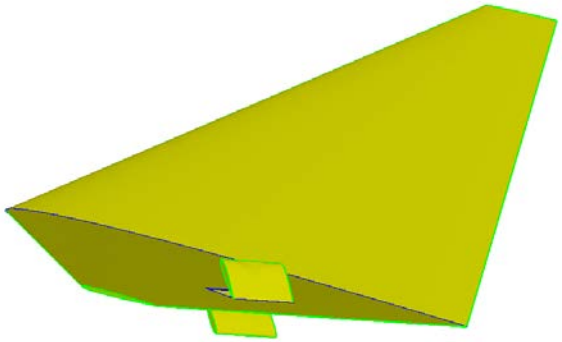


Figure 6a. left-angle=45°, right-angle=0°

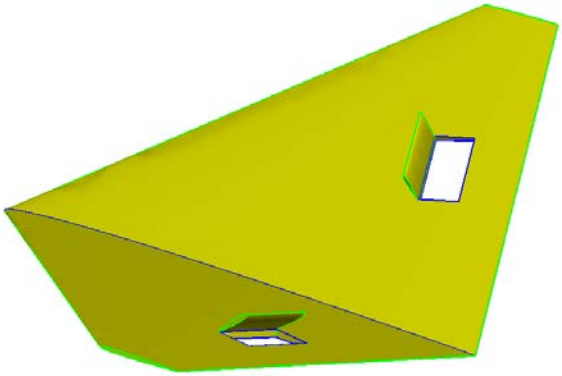


Figure 6b. left-angle=45°, right-angle=90°

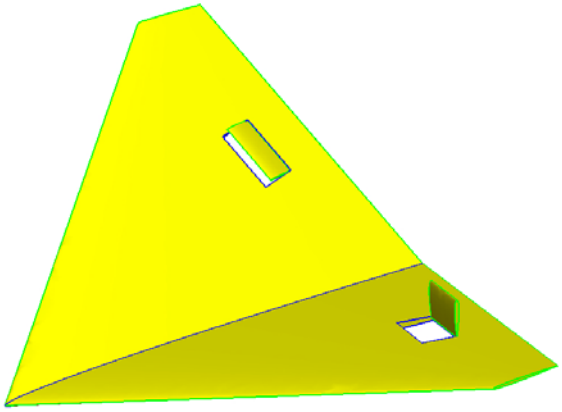


Figure 6c. left-angle=90°, right-angle=15°

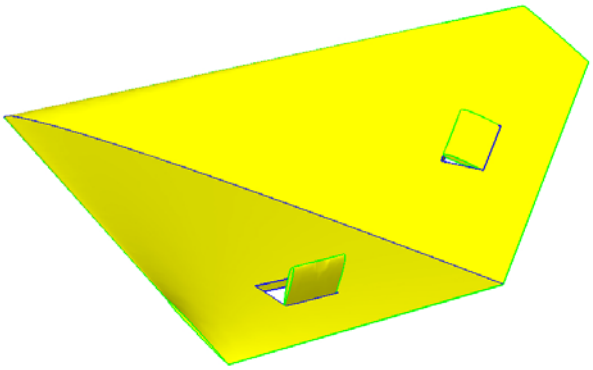


Figure 6d. same as (6c);with twist=25° at mid-chord

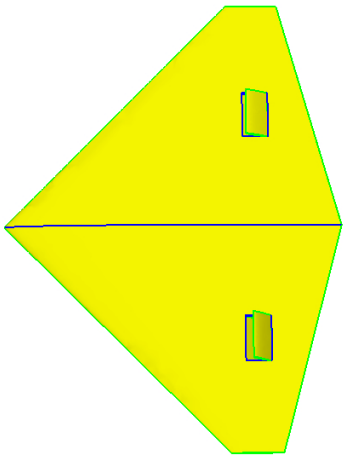


Figure 6e. sweep-angle=-15°, spoiler-orientation=15°

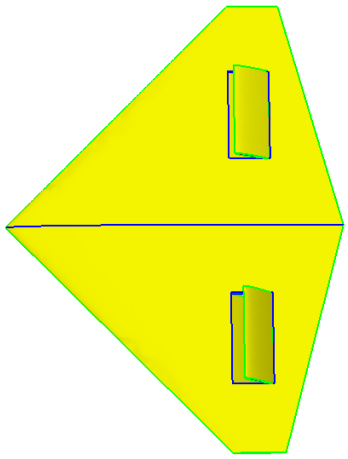


Figure 6f. length=0.8, width=0.4, x=0.2, y=0.3

III. Interfacing Geometry with Analysis Tools

One of the important aspects of making the geometry useful for analysis is meshing. Meshing for an aerodynamic analysis is performed in two stages:

1. Surface meshing over the geometric faces, and
2. Volume meshing inside the field of interest.

Surface meshing is required because the face tessellation used to represent geometry may not have the appropriate resolution to represent the surface for analysis. In this effort, the same geometry must be used for performing multidisciplinary analyses, and these different disciplines may have different requirements for surface meshing. As an example, structural analysis may not need as fine a quality surface mesh as required by fluid analysis. Whereas, volume meshing is required for the aerodynamic analysis to identify and discretize the domain over which the analysis can be performed. Hence, a robust meshing tool is a necessity to satisfy these and other requirements.

To facilitate surface meshing, BAMG, an open-source software is used in conjunction with ESP. The relationship between the two codes is that the edge discretization is performed in ESP and the surface mesh generation is controlled via BAMG's meshing parameter for each face. This provides greater flexibility in generating better quality meshes. As an example, Fig. 7(a) shows the face tessellation provided by ESP when the geometry is initially saved, and Fig. 7(b) shows the surface mesh after using BAMG as meshing tool. The BAMG mesh has the desired resolution.

BAMG is inherently a 2D mesh generator, so the mesh generation is done in parametric space (u,v) instead of real space (x,y,z). Along with the node and edge information, tangents at edges must also be provided to BAMG so that the geometry information is retained while improving the mesh quality. After mesh generation and refinement by BAMG, the parametric mesh is then transformed back into real space. During the surface mesh generation process, the ESP edge discretization is preserved, which makes it easy to transform the surface mesh from parametric space to real space.

Examples of possible resulting surface meshes are shown in Figs. 7(b) through 7(d) at different methods of refinement. Figures 7(c) and 7(d) demonstrate the use of BAMG parameters to improve the surface mesh. Figure 7(b) shows the default mesh (no additional parameters) generated by BAMG using the edge discretization provided by ESP. In the case of Fig. 7(c), the surface mesh was defined using a multiplicative factor (defined as 'coef' in BAMG) of 0.5 for inserting new triangles. In the case of Fig. 7(d), the surface mesh was refined by splitting all existing triangles into four triangles. More options, such as providing curvature information in the form of metric field for BAMG mesh generation are currently being explored and implemented for automation purposes and to improve the resulting mesh quality.

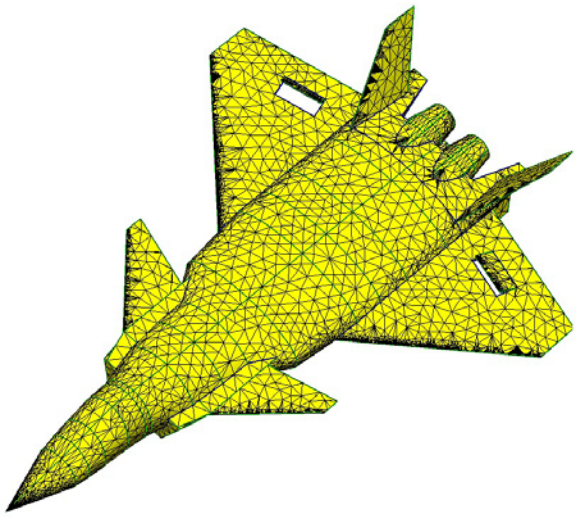


Figure 7a. ESP Face Tesselation

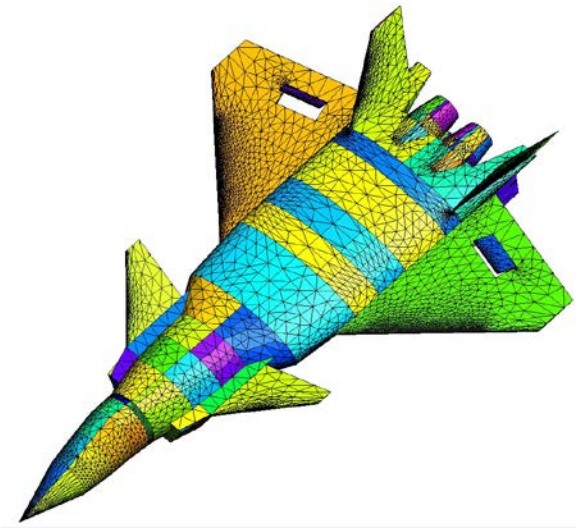


Figure 7b. Initial BAMG Surface mesh

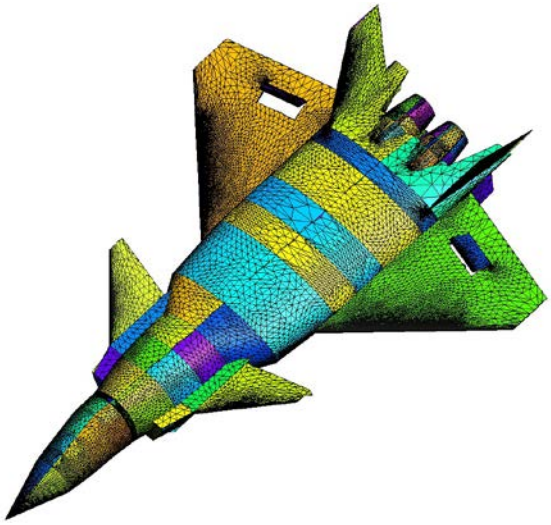


Figure 7c. BAMG mesh refinement: coef=0.5

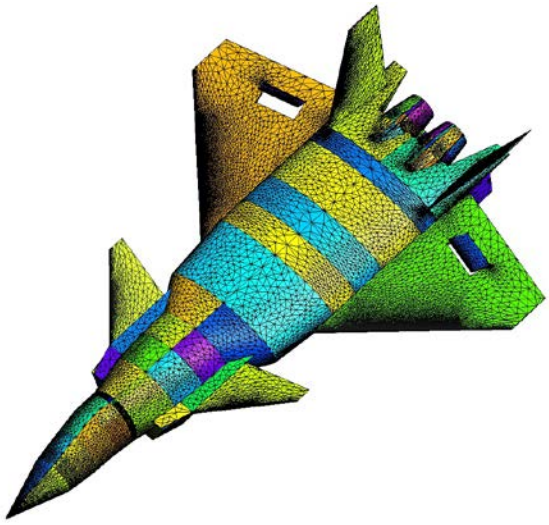


Figure 7d. BAMG mesh refinement: surface triangles split into four triangles

The volume mesh is generated using the open-source unstructured meshing code known as Tetgen. The far field boundaries are located about 4 body lengths from the nose, 5 body lengths from the back, and 4 body lengths on the remaining sides. For this example, the geometry has been simplified by muting the SSDs and inlets. A snapshot of the resulting volume mesh from Tetgen around the aircraft geometry is shown in Fig. 8. The mesh consists of approximately 400 thousand nodes and 2.2 million tetrahedrons.

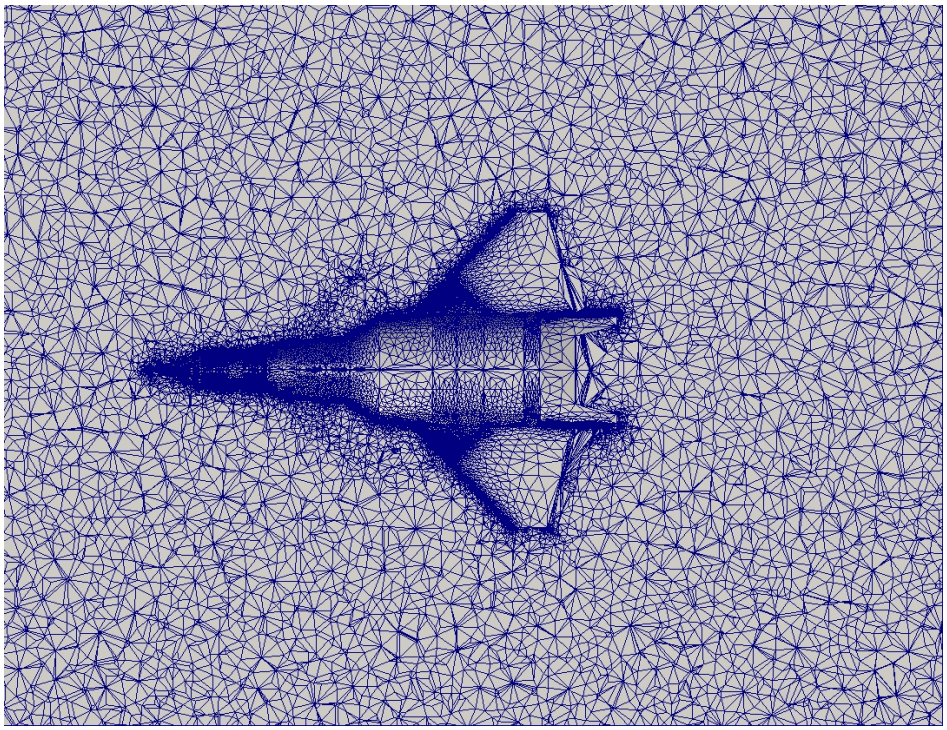


Figure 8: Tetgen volume mesh for the aircraft geometry

Computations on the flow field around the example aircraft geometry that have been carried out using CART3D[1], using different variations of the geometry model, are shown in Fig. 9. The various parametric models were generated by creating a function within ESP to export the required CART3D file format containing the geometry information of the configuration. The configuration introduced in Fig. 2 was explored at multiple Mach numbers and different angles-of-attack and sideslip. No tails were included in these particular studies. One configuration, shown in Fig. 9(a), had no SSDs. The other configurations shown in Figs. 9(b-d) included SSDs. No tails were included because the intent of this example was to test the capability of ESP to aid in the generation of data that could be used to assess tailless flight control rapidly. The goal is to expand the capability of ESP to support CART3D, body fitted unstructured solvers, such as SU[2], and overset codes, such as OVERFLOW,[11] to perform these types of assessments at variable fidelity levels.

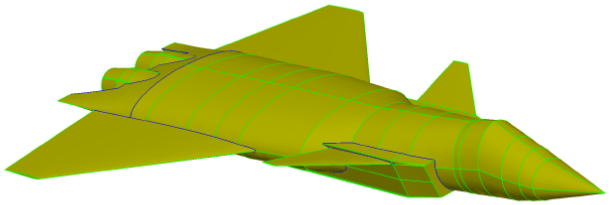


Figure 9a. Tailless Configuration – no SSDs

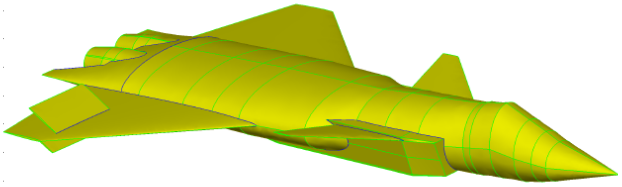


Figure 9b. Tailless Configuration with Single SSD (0° Rotation, 30° Open)

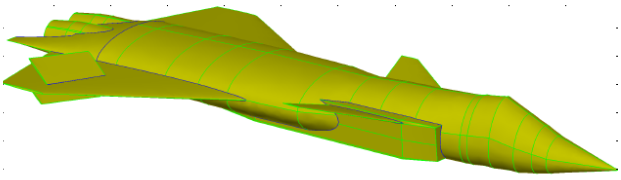


Figure 9c. Tailless Configuration with Single SSD (-20° Rotation, 30° Open)

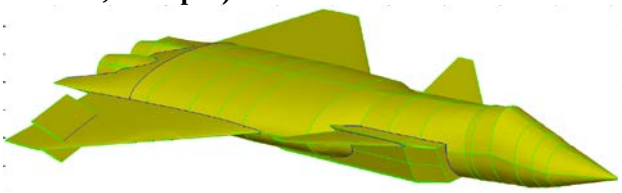


Figure 9d. Tailless Configuration with Single SSD (20° Rotation, 30° Open)

Some example aerodynamic analysis results in the form of yaw and roll coefficients are shown in Figs. 10(a) and 10(b). This data was taken for a flight condition of Mach 0.5 with a side slip angle of 5.0°. This is representative of

a landing approach condition with a cross wind. For the yaw coefficient, the different control effector configurations produce the correct coefficient trends. Likewise, the roll Coefficient trends behave as expected too.

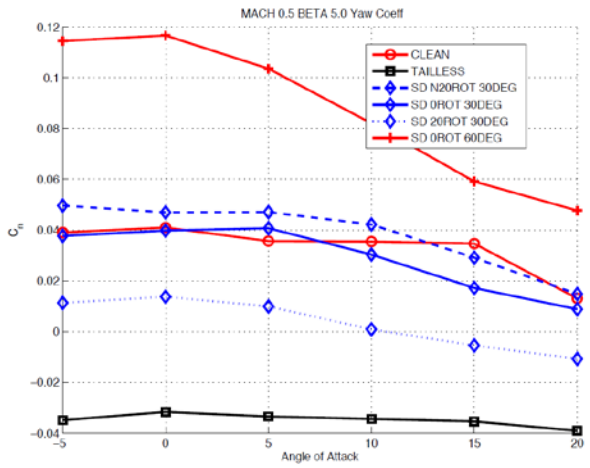


Figure 10a. Yaw Coefficient

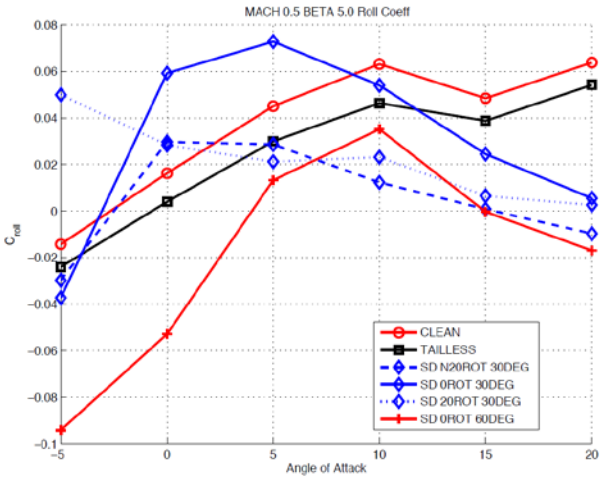


Figure 10b. Roll Coefficient

Further work is being done to expand the analysis capability to support higher-fidelity body fitted unstructured and structured overset aerodynamic analyses. In order to support the body fitted unstructured analysis capability, computations were performed on a cylinder test case using the open-source software SU^2, and the results are shown in Fig. 11. The cylinder geometry was generated and meshed using the process described for the aircraft example previously. This interface is still in its infancy. However, the preliminary results are promising because the results from the flow over a cylinder demonstrate that the interface works (at least at the qualitative level).

Figures 11(a-c) show pressure data on and around the cylinder surface. The surface patches were used for monitoring the convergence (defined in the configuration file). Fig. 11(d) shows streamlines from a point source that lies in the field close to the cylinder. The streamlines are extended in both field directions and are colored by the magnitude of the product of density and velocity. All the analysis results are visualized using the open-source visualization tool Paraview.[12]

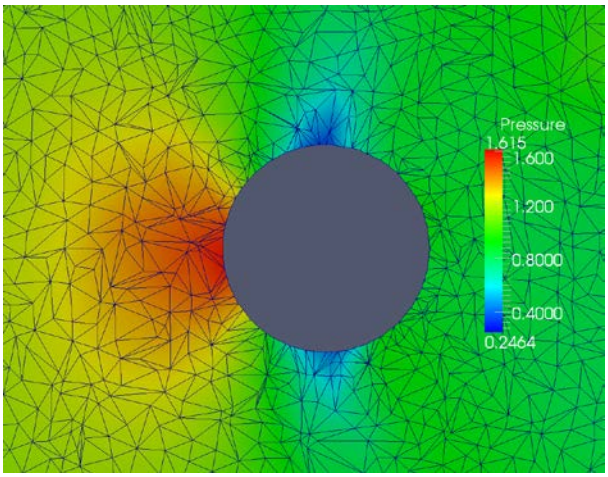


Figure 11a. ESP/SU^2 Analysis: Pressure counters with unstructured mesh showing

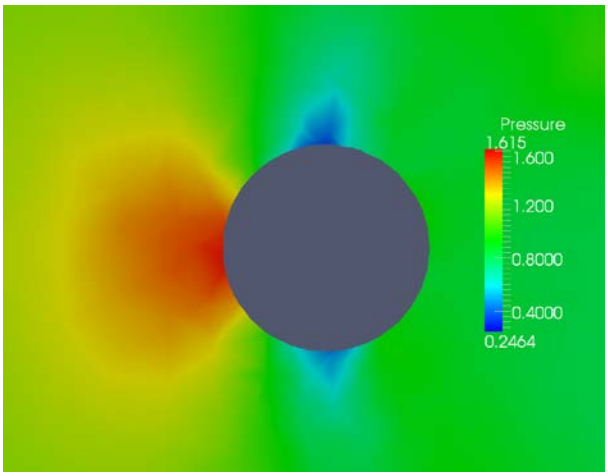


Figure 11b. ESP/SU^2 Analysis: Pressure counters

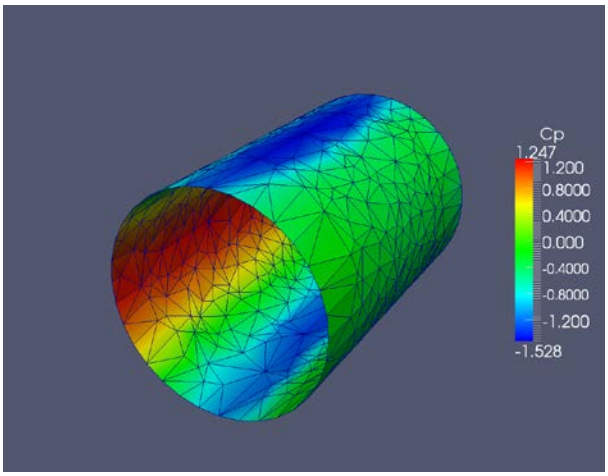


Figure 11c. ESP/SU^2 Analysis: C_p on cylinder surface

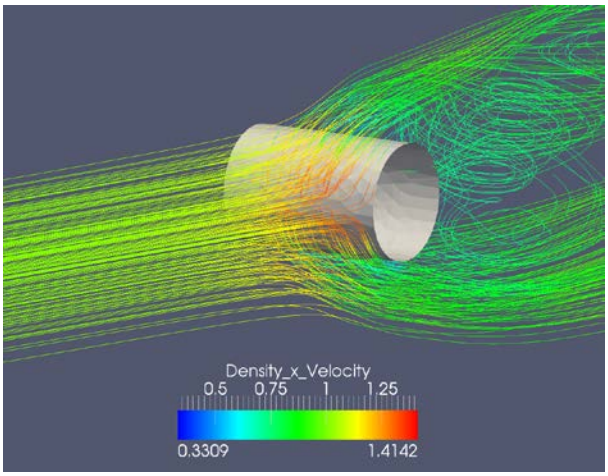


Figure 11d. ESP/SU^2 Analysis: Streamlines over cylinder

IV. Summary

This effort aims to provide a unified geometry model that can be used for multifidelity and multidisciplinary analysis. An aircraft example problem reported herein shows the developing capability and the present status of the effort, especially in terms of automatically generating parameterized geometry models and interfacing them with automatic meshing and analysis. This capability is being used to rapidly asses different tailless control concepts that require nonlinear aerodynamic calculations. Applying ESP to the tailless control problem has already enhanced, and will further enhance, the capability of ESP for multidisciplinary and multifidelity analysis and design optimization. The final goal of this work is rapid access to multifidelity nonlinear aerodynamic and structural assessment.

Acknowledgments

The Primary author would like to thank Jose Camberos and Raymond Kolonay at the United States Air Force Research Laboratory for their support under Contract with Universal Technology Corporation, (Contract number FA8650-10-D-3037). In addition, authors would like to thank Robert Haimes at Massachusetts Institute of Technology, and John F. Dannenhoffer, III at Syracuse University, for various technical discussions, suggestions, and feedback during their visits and various teleconference communications.

References

[1]Alyanak E., Kolonay R., "Efficient Supersonic Air Vehicle Structural Modeling for Conceptual Design", 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2012.

[2]Haimes R., Drela M., "On the Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design", 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2012.

[3]Haimes R., Dannenhoffer J. F., "Control of Boundary Representation Topology in Multidisciplinary Analysis and Design", 48th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2010.

[4]Dannenhoffer J. F., "OpenCSM: An Open-Source Constructive Solid Modeler for MDAO", 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2013.

[5]Haimes R., Dannenhoffer J. F., "The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry", 21st AIAA Computational Fluid Dynamics Conference, 2013.

[6]OpenCASCADE S.A.S., OpenCASCADE, http://www.opencascade.org.

[7]Hetch, F., "BAMG: Bidirectional anisotropic mesh generator", http://www.ann.jussieu.fr/hecht/ftp/bamg, 1998.

[8]Si H., "Tetgen: a quality tetrahedral mesh generator and three-dimensional Delaunay triangulator", 2004.

[9]Aftosmis, M. J., Cart3D, http://people.nas.nasa.gov/~aftosmis/cart3d/, Accessed: May 2013.

[10]Palacios F., Colonno M., Aranake A., Campos A., Copeland S., Economon T., Lonkar A., Lukaczyk T., Taylor T., Alanso J., "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design", 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2013.

[11]OVERFLOW, http://people.nas.nasa.gov/~pulliam/Overflow/Overflow_Manuals.html

[12]Paraview, http://www.paraview.org/paraview/resources/software.php