51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0328

Enabling Rapid Conceptual Design Using Geometry-Based Multi-Fidelity Models in VSP

Joel B. Belben* and Robert A. McDonald†

California Polytechnic State University, San Luis Obispo, CA, 93407

Four types of reduced fidelity or degenerate geometric representations have been defined and implemented in VSP for the purpose bridging the gap between conceptual design and analysis. They are Degenerate Surface, Degenerate Plate, Degenerate Stick, and Degenerate Point, corresponding to three-, two-, one-, and zero-dimensional representations of underlying geometry, respectively. The information contained in these representations was targeted specifically at lifting line theory, vortex lattice, equivalent beam, and equivalent plate methods, but could be used for other analysis techniques. The ability to output this information in both csv and Matlab file formats has been implemented in VSP. Ongoing work seeks to demonstrate the four target analysis techniques in use together with VSP.

Nomenclature

$\bar{x}$	centroid x location
$\bar{z}$	centroid z location
$\hat{\mathbf{c}}$	normalized chord vector from trailing to leading edge
$\hat{n}$	unit normal vector
a	area
b	base of discretized surface section rectangle
d	distance from discretized surface section rectangle centroid to cross-section centroid
I	mass moment of inertia
J	area moment of inertia in global coordinates
J^*	area moment of inertia in section or component coordinates
t	thickness
X	Vector of coordinate locations

Subscripts

$camb$	pertaining to camber line
le	leading edge
n	Normal vector
$plate$	pertaining to Degenerate Plate
te	trailing edge
x	along x direction
xx	about x-axis
yy	about y-axis
z	along z direction
zz	about z-axis

Symbols

ϕ	rotation angle from x-axis
ρ	density

---

[*] Graduate Student, Aerospace Engineering, One Grand Avenue, AIAA Student Member

[†] Associate Professor, Aerospace Engineering, One Grand Avenue, AIAA Senior Member

I. Introduction

Aircraft design is comprised of three main stages: conceptual, preliminary, and detailed.[1] The conceptual design phase is unique among the three in that the exploration of a large design space necessitates designs with large geometric variation.[2] To efficiently produce feasible, novel conceptual designs two things are needed:

1. A method of creating designs with potentially large geometric variation under tight time constraints
2. A way to rapidly analyze these designs using inexpensive analysis techniques to evaluate overall design feasiblity

Vehicle Sketch Pad (VSP) and its predecessors were created specifically to meet the need for rapid geometry creation using high-level design parameters.[2,3] Furthermore, inexpensive yet useful tools exist for aerodynamics and structural analysis (the two main disciplines of concern in the conceptual design phase). What does not exist is a general method of capturing necessary geometric information from a design that enables the use of these and other reduced-fidelity analysis techniques. A reduced-fidelity or *degenerate* geometry is needed as a conduit from design to analysis.

Four target analysis techniques, shown in figure 1, helped guide the definition of degenerate geometry: lifting line theory and vortex lattice for aerodynamics and equivalent beam and equivalent plate theory for structures. Preliminary lift and induced drag distributions are predicted accurately with Prandtl's lifting line theory,[4] and recent additions to the technique have even made it useful for wings with sweep and dihedral.[5] Similarly, reduced-fidelity analysis using Athena Vortex Lattice (AVL) or other vortex lattice methods,[6] equivalent beam theory,[7] and equivalent plate representations[8,9,10,11] have all been shown to provide excellent accuracy at a significantly reduced cost when compared to high-fidelity tools like computational fluid dynamics (CFD).

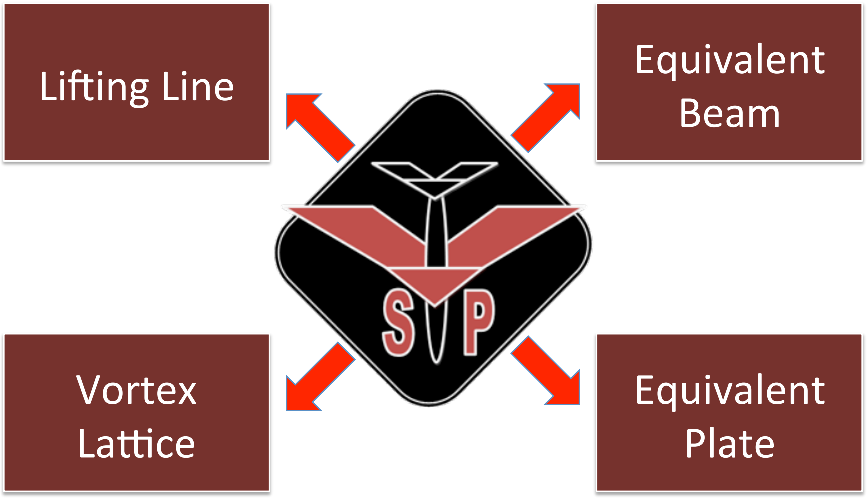


Figure 1. VSP interface to multi-fidelity analysis techniques

Incorporating the ability to write out information from VSP necessary for these and other quick analysis techniques will enable a new design methodology that relies less on historical information and more on the design requirements at hand. It also provides the opportunity to incorporate optimization algorithms into the conceptual design process by adding the ability to automatically interface a scriptable, rapid conceptual design tool like VSP to inexpensive analysis tools. In fact, effort on a VSP plugin for *Model Center*, an optimization environment, is well underway at Phoenix Integration.[12] This will be especially useful for unconventional designs and can even facilitate exploration and learning by engineering students because of the reduced time needed to create a design and asses its feasibility. Breaking down the barriers to the design and analysis process will encourage participation by a wider array of individuals and organizations, resulting in novel ideas and helping push technology forward. The purpose of this paper is to define four levels of degenerate geometry from three- to zero-dimensional, describe their implementation and use in VSP, and demonstrate their usefulness in the design process.

II. Degenerate Geometry Definitions

Aircraft components are broken up into two main categories for the purpose of defining degenerate geometric representations. *Surfaces* are objects such as wings, stabilizers, aerodynamic pylons, etc. They are essentially wing-like objects, lifting or non-lifting (symmetric). *Bodies* encompass everything else from fuselages to propellor spinners and landing gear. The inspiration for this classification system was taken directly from AVL, which categorizes components in precisely this manner.[6] Bodies may play a potentially large role in overall drag, but are not suitable for (basic) aeroelastic studies or to the computation of lift. This classification is also convenient because certain characteristics can be assumed about each geometry type. Surfaces for instance, will have one dimension much smaller than the other two, meaning that qualities like thickness are easily defined.

Figure 2 shows a Cessna 182 model with body components in blue and surface components in red. Notice that the wing and main gear struts are surface types since they are made from aerodynamic, wing-like shapes, whereas the landing gear wheels and pants are body types. Though not shown in this view, the nose gear strut is also a body type, since it was modeled as a right circular cylinder.

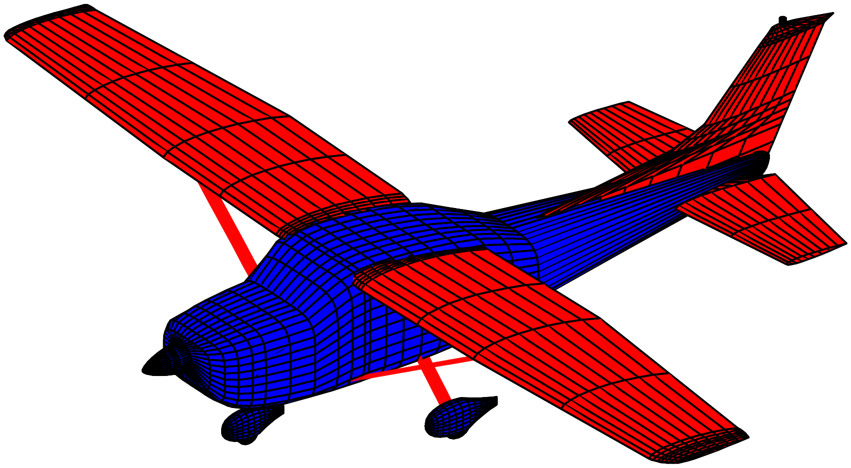


Figure 2. Cessna 182 model showing body (blue) and surface (red) type components.

Four levels of degenerate geometry have been defined. They are, in order of decreasing fidelity: Surface, Plate, Stick, and Point, corresponding to three-, two-, one-, and zero-dimensional representations, respectively. Both surface and body types can be distilled into any one of these four, though their definitions are slightly different depending on the category of the original geometry. The following sections define each of these degenerate geometries, for both surface and body type components. In each of these analyses, it is assumed that a coordinate system is adopted whereby positive x is aft down the fuselage, positive y is out the right wing, and positive z is up.

A. Degenerate Surface

An object's true geometry is three-dimensional, continuous on a macroscopic scale, and may be closely approximated by, but is never entirely amenable to mathematical description. In fact, describing an object mathematically is the first stage in reducing it's fidelity from true geometry to some tractable characterization. Doing so generally requires selecting points that comprise the object and connecting them in a piecewise fashion with curves of a desired order, the simplest of which are straight lines. These control points then, can be thought of as a reduced-fidelity representation of true, underlying geometry. In fact a Degenerate Surface has been defined such that it contains a collection of these control points. Connected together, they form a surface which approximates the original object.

Figure 3 shows this discretization on a Cirrus SR22, with each component shown in a different color. Note that the points have been grouped into cross-sections along each component, and though not obvious in the figure, each cross-section contains the same number of points within any component. The level of fidelity (or accuracy of representation) of any cross-section is limited only by how many control points are used, with the model approaching the actual object as the number of cross-sections and control points becomes infinite.

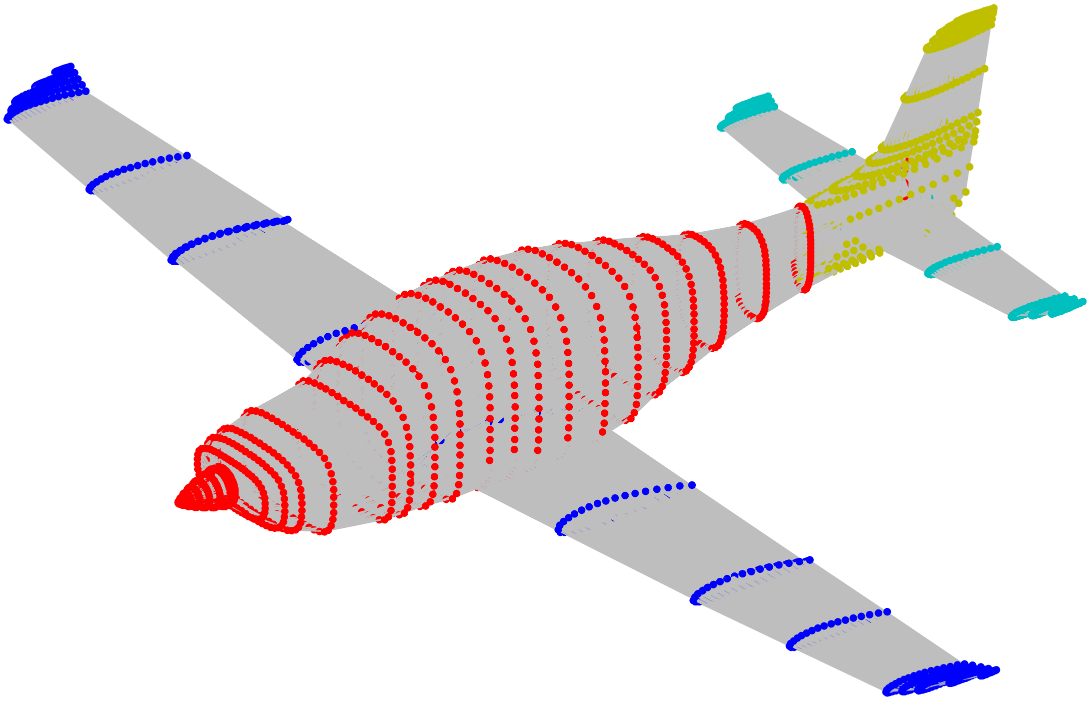


Figure 3. Cirrus SR22 model, showing Degenerate Surface representation.

A natural question is what form this collection of control points takes. The answer is a simple vector of coordinates ordered by cross-section. If a component has p cross-sections and q points per cross-section, then there are $m = p \times q$ control points and the vector of these control points is

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]^\intercal \tag{1}$$

where each $\mathbf{x}_i$ is a cartesian coordinate triplet (x_i, y_i, z_i). The number of cross-sections and points per cross-section are included as a means of effectively using this information. Degenerate Surface also provides outward surface normal vectors in the form

$$\mathbf{X}_n = [\hat{n}_1, \hat{n}_2, \cdots, \hat{n}_r]^\intercal \tag{2}$$

where each $\hat{n}_i$ is a normalized cartesian coordinate triplet (x_i, y_i, z_i) describing the outward-facing surface normal direction. Since a point has no single outward direction, normal vectors are defined using surrounding points. If control points within a cross-section are indexed by i and cross-sections by j, then a normal vector $n_{ij} = t_1 \times t_2$, where $t_1 = x_{i,j+1} - x_{ij}$ and $t_2 = x_{i+1,j} - x_{ij}$ and $\hat{n}_{ij} = n_{ij}/\|n_{ij}\|$, as shown in figure 4. Note that for each cross-section, there will be one less normal vector than control point, and the last cross-section will have no normal vectors. The length of $\mathbf{X}_n$ is $r = (p-1)(q-1)$. Note also that though degenerate geometries are categorized as either surface or body types, both types can be described with the same Degenerate Surface definition.

B. Degenerate Plate

The next step in fidelity reduction is to represent a three-dimensional object as two-dimensional. Since surface type components have one dimension much smaller than the other two, it's quite natural to collapse them down on that dimension. In fact, this two-dimensional, plate representation was inspired by both equivalent plate structural analysis and vortex lattice aerodynamic analysis, which are primarily concerned with wings and wing-like components[6] [11]. For this reason Degenerate Plate's definition relies on airfoil nomenclature

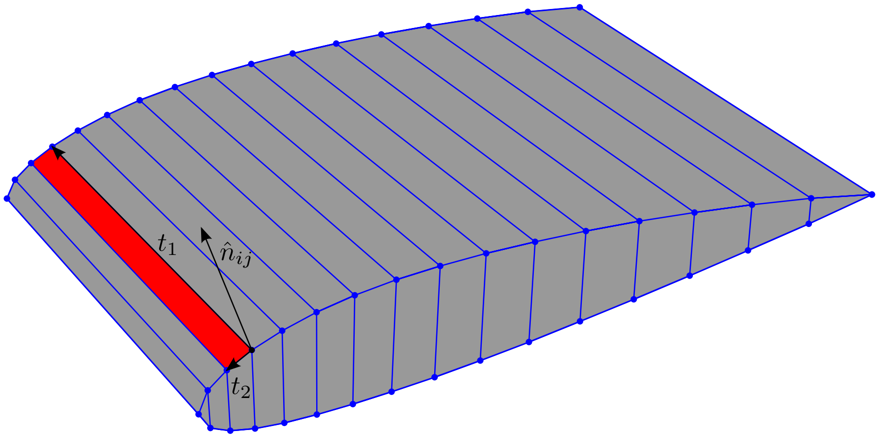


Figure 4. An example Degenerate Surface normal vector on a wing section.

and general geometry. Unlike Degenerate Surfaces, Degenerate Plates need separate definitions for surface and body type components. For simplicity, a surface definition is provided first.

To create a Degenerate Plate, an object is first discretized into a series of cross-sections, which are each represented by a number of coordinate points, essentially a Degenerate Surface without normal vectors. These points are then collapsed down to a planar representation as shown in figure 5.

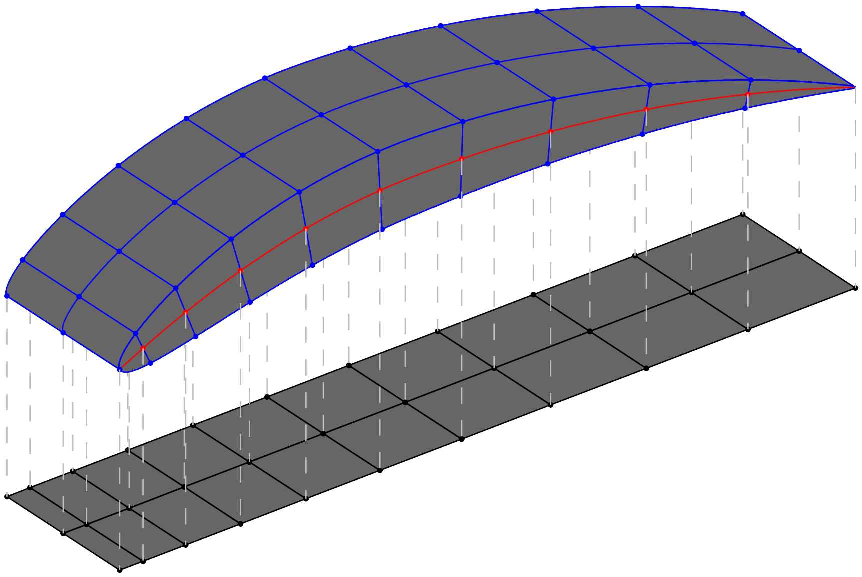


Figure 5. Mapping between discretized surface points and Degenerate Plate points for a wing section.

Figure 6 shows details of how these points are mapped from a single cross-section to a plate. First, the midpoints between corresponding upper and lower nodes are calculated via

$$\mathbf{X}_{camb} = \frac{1}{2}\left(\mathbf{X}_{top} + \mathbf{X}_{bottom}\right) \tag{3}$$

where the $\mathbf{X}$s are vectors of coordinate points (x, y, z) defining the upper and lower surfaces. These midpoints form the camber line shown in red. The computed camber points are then projected onto the sectional chord line using vector projection. If x_{te} and x_{le} are the trailing and leading edge coordinates, respectively, then a normalized vector along the chord pointing from the trailing edge to the leading edge is given by

$$\hat{\mathbf{c}} = \frac{x_{le} - x_{te}}{\|x_{le} - x_{te}\|} \tag{4}$$

If a vector from the trailing edge to a camber line point a is given by $\mathbf{a}$, then the Degenerate Plate point b corresponding to a is given by

$$b = x_{te} + \hat{\mathbf{c}} \times (\mathbf{a} \cdot \hat{\mathbf{c}}) \tag{5}$$

At each node b, Degenerate Plate also reports the vertical distance to a,

$$\Delta z_{camb} = \|a - b\| \tag{6}$$

thickness of the original section t and the camber line normal vector $\hat{n}_{camb}$. Additionally, for each cross-section a plate normal vector $\hat{n}_{plate}$ is given, which gives the original orientation of the cross-section, and defines the direction from the plate points b to the chord points a.

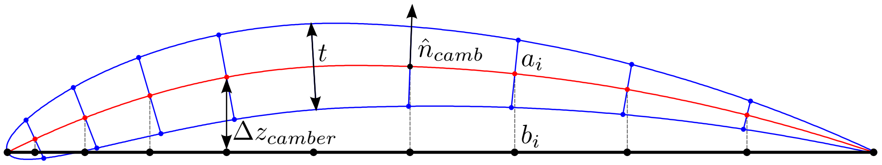


Figure 6. Degenerate Plate attribute definition using an airfoil section.

Having sufficiently defined a Degenerate Plate representation of wing-like components, a question arises concerning what this definition is for something like a fuselage. Defining a general plate orientation to represent thick bodies is decidedly difficult. If an object is axisymmetric, then any section which bisects it into two symmetric pieces would make an appropriate plate. However, the vast majority of aircraft body parts are not axisymmetric and so a different approach is necessary. Since it is virtually impossible to state, in a general sense the least dominant dimension for arbitrary body geometry, it was decided that collapsing a part along two separate dimensions was the best alternative. This means that Degenerate Plates composed of body geometry actually contain two plate objects. This is easiest to see using a right circular cylinder as shown in figure 7. Note that in this example the plates are orthogonal, but this is not always the case. The two plates are defined such that they equally divide the number of discretized nodes in each cross-section. This means that if the nodes are unequally distributed (i.e more on the left than right half, etc.) the plates will assume non-orthogonal orientations. Additionally, since this is done on a per cross-section basis the plates' locations can vary along a body, meaning that its Degenerate Plate representation is not necessarily planar in a cartesian coordinate system. Aside from creating two Degenerate Plates from each component, the definitions of thickness, distance to camber lines (note that there are two for body type components), etc. are all analogous to surface type Degenerate Plates. Figure 8 shows a Cessna 182 Degenerate Plate model 3 view, with each component shown in a different color. All components are surface types and hence collapse down to single plates, with the exception of the fuselage which is represented by two plates.

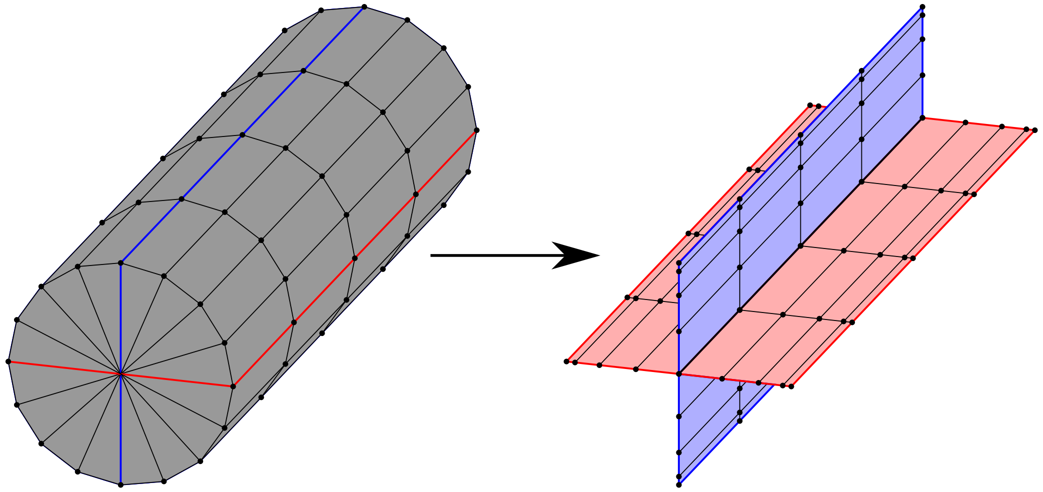


Figure 7. Transformation from body component to Degenerate Plate using a right circular cylinder.

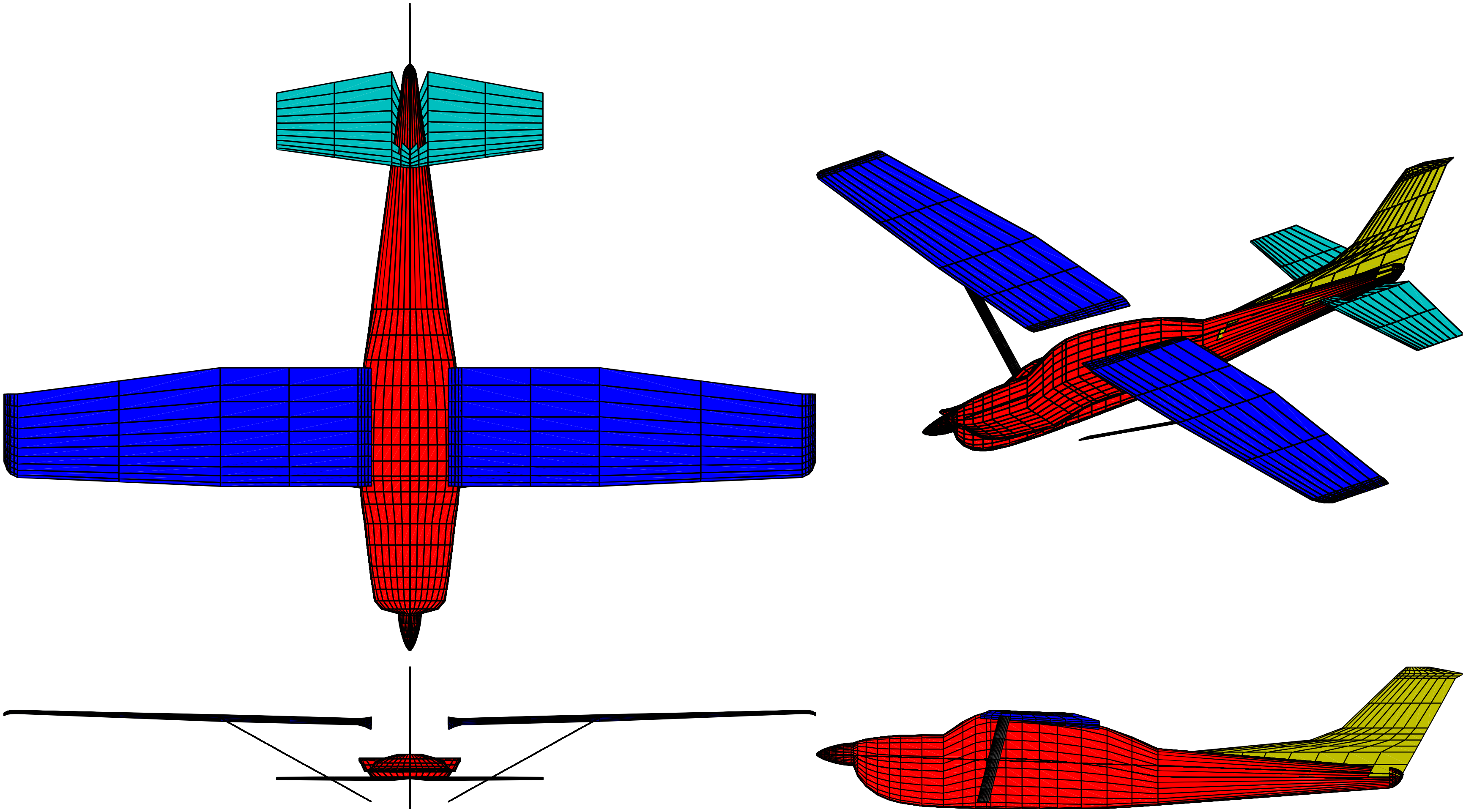


Figure 8. Degenerate Plate model of a Cessna 182

C. Degenerate Stick

Degenerate Stick reduces fidelity further from Degenerate Plate, creating a one-dimensional representation, where each point on the Degenerate Stick corresponds to a cross-sectional slice of the actual geometry. Like Degenerate Plate, Degenerate Stick relies on airfoil nomenclature, but has separate definitions for surfaces and bodies. Once again, the surface definition is presented first, followed by extension to the body definition.

If a Degenerate Plate is a Degenerate Surface collapsed to a plane, then a Degenerate Stick is a Degenerate Plate collapsed to a line. To create a Degenerate Stick, an object is first discretized into cross-sections, with each one corresponding to a Degenerate Stick node. Figure 9 shows these nodes connected together (red line) along with the original wing. Though the nodes and connecting line shown are an easy way to visualize a Degenerate Stick, the points that define it are actually the leading and trailing edge points from the original component (shown in blue). Degenerate Sticks also report maximum thickness to chord, max thickness location in percent chord, chord length, cross-sectional area, an area normal vector, and top and bottom perimeters, as shown in figure 9. Not shown is the quarter chord sweep angle which is defined between adjacent cross-sections, so that there is one less sweep angle than Degenerate Stick node.

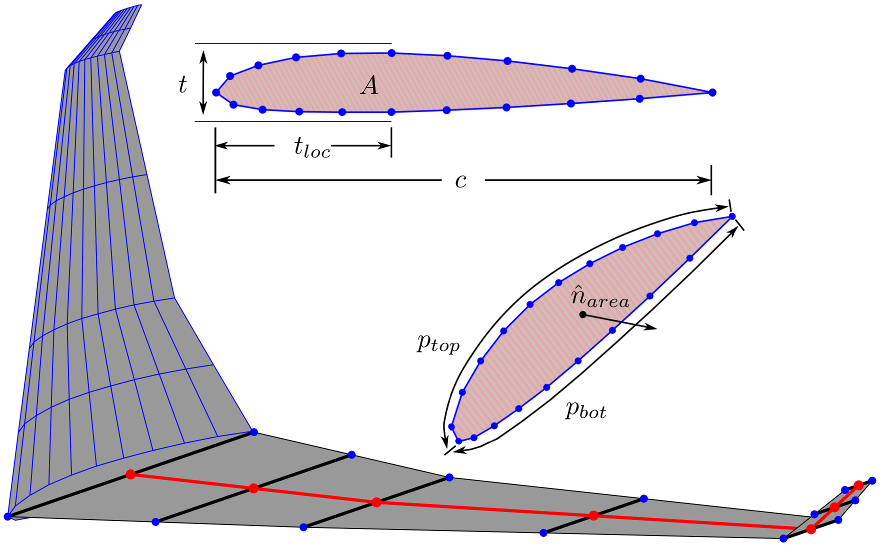


Figure 9. Degenerate Stick model of Boeing 747 Wing.

Of interest in equivalent beam structural analysis are sectional moments of inertia, specifically those resisting lift and drag and a torsional moment of inertia. These can be provided in a general manner for both solid and thin-walled "shell" sections without knowing the units or wall thickness properties. Since creating a Degenerate Stick requires discretization of a component into cross-sections composed of points, each cross-section can be treated as a polygon defined by n points. Research by Steger[13] resulted in formulae for moments of arbitrary order for polygons. Those for area moments about the origin are given by

$$J_{xx} = \frac{1}{12}\sum_{i=1}^{n}(x_{i-1}z_i - x_i z_{i-1})(zi - 1^2 + z_{i-1}z_i + z_i^2) \tag{7}$$

$$J_{zz} = \frac{1}{12}\sum_{i=1}^{n}(x_{i-1}z_i - x_i z_{i-1})(xi - 1^2 + x_{i-1}x_i + x_i^2) \tag{8}$$

The parallel axis theorem is employed in order to get moments of inertia about cross-section centroid:

$$J^*_{xx} = J_{xx} - a\bar{z}^2 \tag{9}$$

$$J^*_{zz} = J_{zz} - a\bar{x}^2 \tag{10}$$

where $(\bar{x}, \bar{z})$ is the cross-sectional centroid coordinates and a the area. These can found via

$$a = \frac{1}{2} \sum_{i=1}^{n} x_{i-1} z_i - x_i z_{i-1} \tag{11}$$

$$\bar{x} = \frac{1}{6a} \sum_{i=1}^{n} (x_{i-1} z_i - x_i z_{i-1})(x_{i-1} + x_i) \tag{12}$$

$$\bar{z} = \frac{1}{6a} \sum_{i=1}^{n} (x_{i-1} z_i - x_i z_{i-1})(z_{i-1} + z_i) \tag{13}$$

It is assumed that component coordinates are given such that lift acts in the z-direction, and drag in the x-direction. Equations (9) and (10) then correspond to resistance to lift and drag, respectively. The resistance to torsion J^*_{yy} is about the y-axis and is simply the sum of J^*_{xx} and J^*_{zz}. All solid cross-section inertias are given in units to the fourth power.

If a cross-section is instead treated as a shell with small thickness, then each internodal line segment can be treated as a rectangle. Figure 10 shows an airfoil section broken up into rectangles, with thickness exaggerated to show detail. If the length of each segment is b and the thickness t, then the moments of inertia about the centroid c can be found via

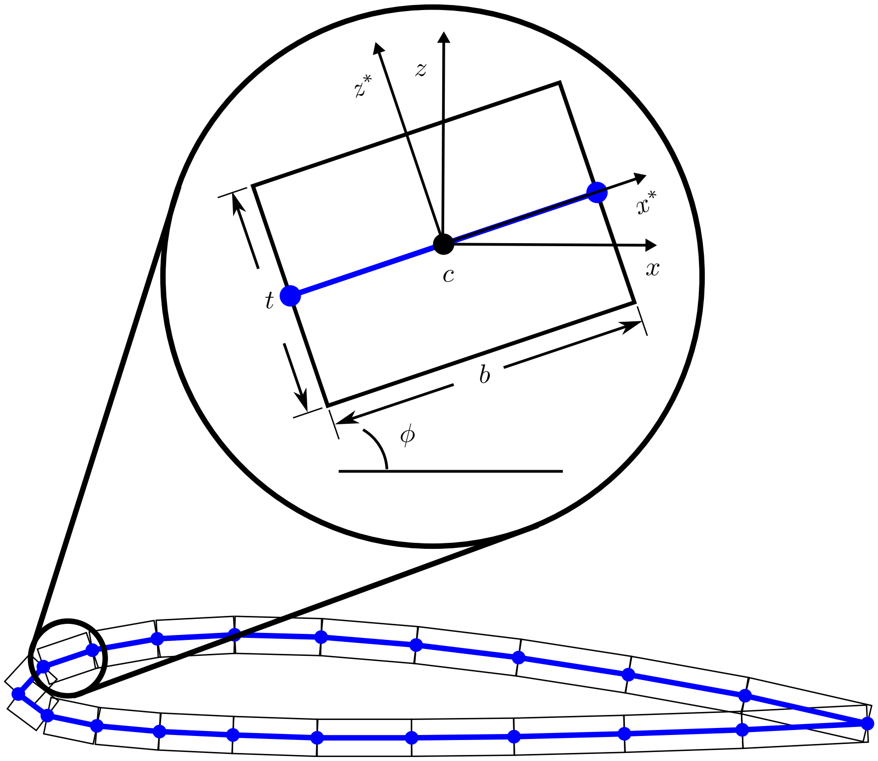


Figure 10. Shell representation of an airfoil section using rectangles to model thickness.

$$J^*_{xx} = \frac{bt^3}{12} \tag{14}$$

$$J^*_{zz} = \frac{b^3 t}{12} \tag{15}$$

$$J^*_{xz} = 0 \tag{16}$$

Rotating these inertias by ϕ so that they align with the global coordinate system and applying the parallel axis theorem, gives the contribution to cross-sectional inertia of each discretized segment

$$J_{xx} = \frac{J_{xx}^* + J_{zz}^*}{2} + \frac{J_{xx}^* - J_{zz}^*}{2}\cos(2\phi) - J_{xz}^*\sin(2\phi) + btd_x^2 \tag{17}$$

$$J_{zz} = \frac{J_{xx}^* + J_{zz}^*}{2} - \frac{J_{xx}^* - J_{zz}^*}{2}\cos(2\phi) + J_{xz}^*\sin(2\phi) + btd_z^2 \tag{18}$$

where d_x and d_z are the distances from c to the cross-section centroid along the x-axis and z-axis, respectively. Substituting equations (14), (15), and (16) into (17) and (18) yields equations of the form

$$J_{xx} = at^3 + bt \tag{19}$$

$$J_{zz} = ct^3 + dt \tag{20}$$

where the coefficients a, b, c, and d are given by

$$a = \frac{b}{24}\left[1 + \cos\left(2\phi\right)\right] \qquad c = \frac{b}{24}\left[1 - \cos\left(2\phi\right)\right]$$

$$b = \frac{b^3}{24}\left[1 - \cos\left(2\phi\right)\right] + bd_x^2 \qquad d = \frac{b^3}{24}\left[1 + \cos\left(2\phi\right)\right] + bd_z^2$$

The total cross-sectional inertia is simply the sum of the contributions of each segment and is given in the form

$$J_{xx,tot} = At^3 + Bt \tag{21}$$

$$J_{zz,tot} = Ct^3 + Dt \tag{22}$$

where the coefficients are now the sum of the respective coefficients from each segment

$$A = \sum_{i=1}^{n} a_i \qquad C = \sum_{i=1}^{n} c_i$$

$$B = \sum_{i=1}^{n} b_i \qquad D = \sum_{i=1}^{n} d_i$$

Recall that since these are aligned with the global coordinate axes and lift is assumed to act in the z-direction, $J_{xx,tot}$ and $J_{zz,tot}$ are cross-sectional resistances to bending due to lift and drag, respectively. The resistance to torsion is $J_{yy,tot}$ and is the sum of $J_{xx,tot}$ and $J_{zz,tot}$.

Degenerate Stick reports these four coefficients A, B, C, and D for each cross-section so that shell inertia is defined as a function of thickness. Note that approximating the surface as a series of rectangles relies on a thin wall assumption so that the error due to overlapping segments is small. Most shell structures for aerospace applications fit this criteria.

As mentioned previously, the definition of Degenerate Stick needs some extension to deal with body type components. In a similar manner to Degenerate Plate, this is accomplished by collapsing underlying geometry down along two separate (nominally orthogonal) directions and reporting two sets of information for each cross-section. This is again most easily shown using a right circular cylinder. Figure 11 shows how a body component is transformed into a Degenerate Stick. The "leading edge" points are shown in blue, while the "trailing edge" points are shown in red. Note that the Degenerate stick shown in black with black and teal nodes actually has two sets of data reported at each node, or there are two Degenerate Sticks overlaying one another.

D. Degenerate Point

The final step in fidelity reduction is to treat a component as zero-dimensional, or a geometric point. Degenerate Point does this for shell or solid components, actually reporting the center of gravity location for each of these two cases. Additional information that Degenerate Point contains is the component volume, area, wetted volume, wetted area, and mass moments of inertia for a solid and shell. The wetted properties make Degenerate Point unique among the degenerate representations, in that it is the only one that relies

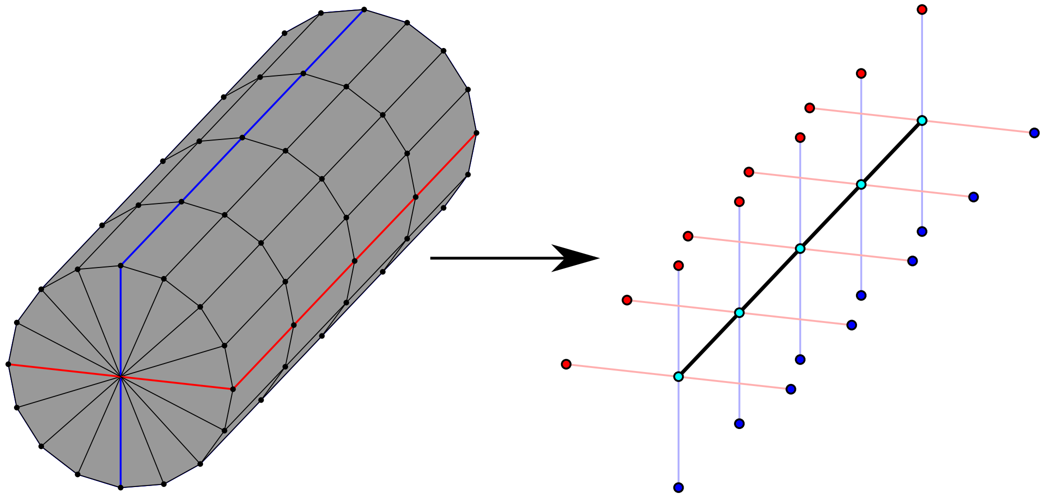


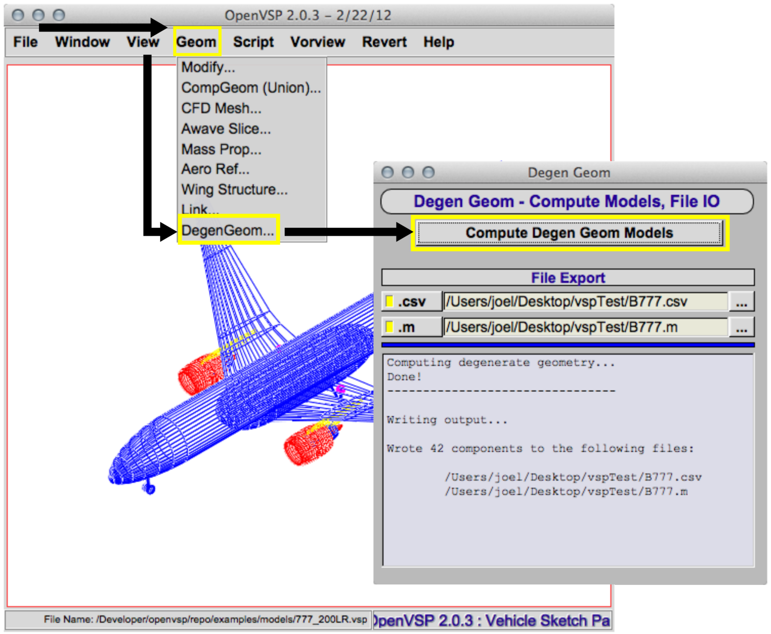
Figure 11. Transformation from body component to Degenerate Stick using a right circular cylinder.

on other components for information. External component geometry is needed to find what portions of the area and volume of the component of interest are intersected.

Six inertia values are given for both a solid and shell representation of each component: I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}, with the products of inertia assumed symmetric (i.e. $I_{xy} = I_{yx}$). For solid components, these are given per density and for shell components per surface density. To get moments of inertia, simply multiply by density ρ, or by density and thickness, ρ and t depending on which set of inertias, solid of shell, are desired.

III. Implementation in VSP

All four Degenerate Geometry types have been implemented in VSP along with the ability to write this information out to file. Figure 12 shows how to access these capabilities. Like other geometric information



.csv /Users/joel/Desktop/vspTest/B777.csv

.m /Users/joel/Desktop/vspTest/B777.m

```
Computing degenerate geometry...
Done!
-------------------------------

Writing output...

Wrote 42 components to the following files:

        /Users/joel/Desktop/vspTest/B777.csv
        /Users/joel/Desktop/vspTest/B777.m
```

File Name: /Developer/openvsp/repo/examples/models/777_200LR.vsp

Figure 12. Sequence for creating Degenerate Geometry in VSP.

and conversion capabilities, Degenerate Geometry is under the Geom menu. The Degen Geom user interface (UI) allows a designer to compute all four Degenerate Geometry types and to output them in one or both of a comma-separated value (csv) file or a Matlab script (m-file) that loads all relevant information into a Matlab structure when run.

VSP's main class is `Aircraft` which among other things, holds references to all geometric components. `Aircraft` also holds an instance of the `DegenGeom` class, the container for all four Degenerate Geometry components, if one has been created. `DegenGeom` is composed of four C-structs corresponding to the four Degenerate Geometry types as shown in figure 13. `DegenGeom` also holds an enumeration which tags a component as either a body or surface type. Currently, the only surface type in VSP is `MS WING`. All other components are body types.

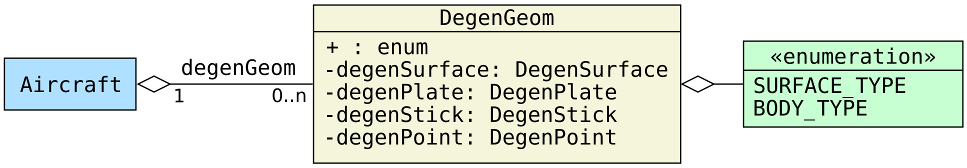


Figure 13. Composition of the DegenGeom class in VSP.

Each aircraft component subclasses `Geom`, the top-level geometry class. VSP stores this component geometry in instances of `Xsec_surf` as two-dimensional arrays of point locations grouped into cross-sections. In other words every component at its core is simply a collection of point locations stored in an instance of `Xsec_surf`. The methods to create all types of Degenerate Geometry except Degenerate Point belong to `Xsec_surf`. When the user tells VSP to create Degenerate Geometry, the function calls are passed all the way down to `Xsec_surf`, which returns an instance of `DegenGeom` back up to `Aircraft`. This process is shown in figure 14. After Degenerate Surface, Degenerate Plate, and Degenerate Stick have been created and their containing instance of `DegenGeom` passed back to `Aircraft`, `Aircraft` creates Degenerate Point and stores this new `DegenGeom`.

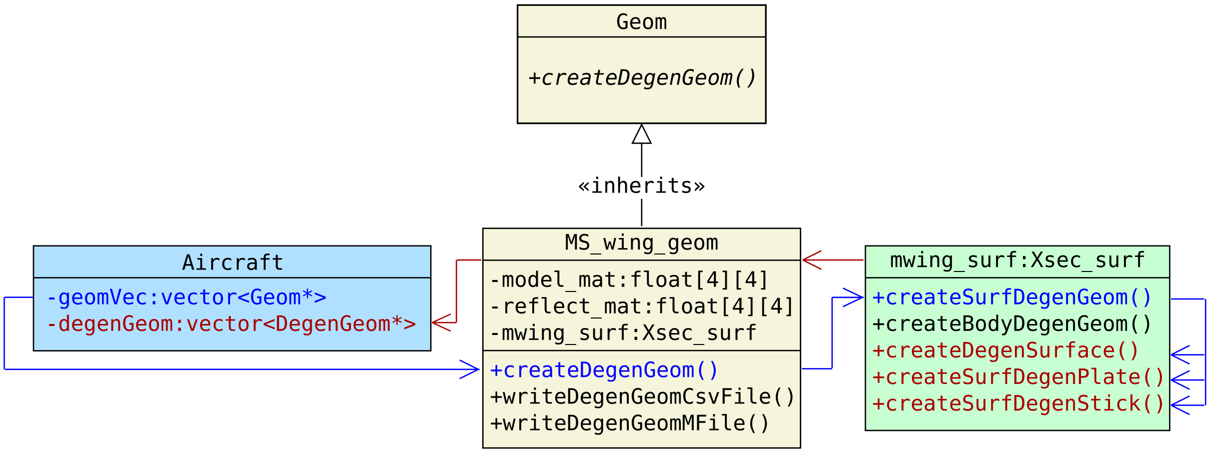


Figure 14. VSP's internal Degenerate Geometry creation process.

It is important to understand this code structure, why it is useful, and what its shortcomings are. The benefit of creating Degenerate Geometry in this way is flexibility. By choosing this structure, future user-defined parts can be distilled into degenerate representations with minimal impact to the existing code. This is because the geometry creation methods (and write to file methods) are all written at a generic, collection of points level. The only information added at higher levels is component name and what type (surface of body) it is. For this reason, future components only need to specify these two parameters (and comply with VSP convention, which they presumably would already need to do). Degenerate Point, as previously mentioned, is treated a little differently. The reason is that all components need to be accessible in order to calculate wetted volume and area, and so the methods to create Degenerate Point are implemented at the `Aircraft` level. This current setup is not the most efficient or flexible, but it is the only feasible way, barring large changes to the VSP code structure.

The motivation for creating Degenerate Geometry was to quickly communicate information to external analysis tools. VSP outputs degenerate geometric information in two ways:

1. A comma-separated value file
2. A Matlab script that loads information into a structure when run

The first fulfills two roles: an easily parsed text file that can be read in and translated to an input file for any number of analysis tools, and a ready-made Excel file, complete with comments. The second is equally useful, as Matlab has become a popular choice for analysis both in industry and academia. This means that VSP output is available immediately for analysis to those designers whose toolsets are written in Matlab. Together, VSP's design capabilities and degenerate geometric information output drastically reduce the time from aircraft design to analysis.

IV. Conclusion

Four types of reduced-fidelity geometric representations have been defined: Degenerate Surface, Degenerate Plate, Degenerate Stick, Degenerate Point in response to a need for bridging the gap between conceptual design and analysis. Work is ongoing in demonstrating the time savings and feasibility of using degenerate geometries with the four target analysis techniques previously mentioned. Though, the degenerate representations were defined with these target analyses in mind, the intention is that the information they contain is suitable for a wide range of low- to mid-fidelity analysis tools.

The ability to compute these representations and write the information to file has been implemented in VSP. Output types are either csv file or Matlab script. With this new feature, the ability to output carefully selected, analysis-ready information about any component, VSP becomes an even more powerful tool for aircraft conceptual design.

An interface between geometry and analysis tools also makes incorporating optimization into the design cycle feasible, inexpensive, and relatively easy. This seamless transition between design and analysis has the potential to facilitate full exploration of design spaces where it once was too time-consuming. Breaking down the barriers to design and analysis encourages more individuals to contribute to the field and facilitates conceptual design at the academic level, spawning new and novel ideas and pushing technology forward.

References

[1] Nicolai, L. M., Carichner, G., and American Institute of Aeronautics and Astronautics., *Fundamentals of aircraft and airship design. Volume I, Aircraft design.*, AIAA, Reston, Va, 2010.

[2] Hahn, A., "Vehicle sketch pad: a parametric geometry modeler for conceptual aircraft design," *Proceedings of the 48th AIAA Aerospace Sciences ...*, , No. January, 2010, pp. 1–11.

[3] Gloudemans, J., Davis, P., and Gelhausen, P., "A Rapid Geometry Modeler for Conceptual Aircraft," *Aerospace Sciences Meeting ...*, , No. January, 1996.

[4] Anderson, J. D., *Fundamentals of aerodynamics*, McGraw-Hill, New York, 2011.

[5] Phillips, W. F. and Snyder, D. O., "Modern Adaptation of Prandtl's Classic Lifting-Line Theory," *Journal of Aircraft*, Vol. 37, No. 4, July 2000, pp. 662–670.

[6] Drela, M. and Youngren, H., "AVL 3.30 User Primer," Aug. 2010.

[7] Lee, U., "Equivalent continuum beam-rod models of aircraft wing structures for aeroelastic analysis," *Archive Set 536*, Meeting Paper Archive, American Institute of Aeronautics and Astronautics, Jan. 1963.

[8] Giles, G., "Equivalent plate analysis of aircraft wing box structures with general planform geometry," *Archive Set 293*, Meeting Paper Archive, American Institute of Aeronautics and Astronautics, Jan. 1963.

[9] Giles, G., "Equivalent plate analysis of aircraft wing box structures with general planform geometry," *Journal of Aircraft*, , No. March, 2012.

[10] Giles, G., "Further Generalization of an Equivalent Plate Representation for Aircraft Structural Analysis," *Journal of Aircraft*, , No. February, 1989.

[11] Giles, G., "Equivalent plate modeling for conceptual design of aircraft wing structures," Los Angeles, CA, 1995.

[12] Ko, A., "The OpenVSP ModelCenter Plug-In," Presented at the First OpenVSP Workshop, California Polytechnic State University, San Luis Obispo, CA, 2012.

[13] Steger, C., "On the calculation of arbitrary moments of polygons," *... Munchen, Germany, Tech. Rep. FGBV-96 ...*, , No. October, 1996.