A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimization

Daniel M. Fudge* and David W. Zingg†
University of Toronto Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

and

Robert Haimes‡
Massachusetts Institute of Technology, Aerospace Computational Design Laboratory, Cambridge, Massachusetts, USA, 02139

The performance of an aerodynamic shape optimization routine is greatly dependent on its geometry control system. This system must accurately parameterize the initial geometry and generate a flexible set of design variables for the optimization cycle. It must also generate new instances of the geometry based on the changes to the design variables dictated by the optimization routine. In response to changes in the geometry, it is also desirable to generate a new surface grid with the same topology as the original grid. This new surface grid can be used to perturb the associated volume grid. This paper presents two geometry control systems, a CAD-free system, and a CATIA V5 CAD-based system. The two systems provide practical tools for aerodynamic optimization. They also provide a basis for comparing CAD-free and CAD-based systems and understanding additional issues that need to be addressed in order to develop reliable optimization systems.

Nomenclature

API	=	Application Program Interface
CAD	=	Computer Aided Design
CAPRI	=	Computational Analysis Programming Interface
CFD	=	Computational Fluid Dynamics
NURBS	=	Non-Uniform Rational B-Splines
u, v	=	Two-Dimensional Parametric Coordinates

I. Introduction

A. Background

THE motivation for the present work is the extension of the Newton-Krylov algorithm for two-dimensional aerodynamic optimization developed by Nemec and Zingg[1,2,3] to optimization of three-dimensional geometries, such as wings and wing-fuselage configurations. The two-dimensional algorithm, which utilizes the discrete adjoint method, has been applied to the design of single- and multi-element airfoils. Since the range of geometries and topologies of interest in two dimensions is quite narrow, this can be accomplished with a relatively simple geometry parameterization and a similarly straightforward grid perturbation technique. In three dimensions, the geometry representation is much more complex, having to deal with a wide range of topologies including features such as wing-body junctions. Furthermore, in three dimensions, it becomes more important to have a geometry representation that is compatible with a computer-aided-design (CAD) representation, i.e. it can be initialized from CAD geometry and returned to CAD format after optimization.

* Graduate Student, Computational Aerodynamics, daniel@oddjob.utias.utoronto.ca.
† Professor, Senior Canada Research Chair in Computational Aerodynamics, 2004 Guggenheim Fellow, dwz@oddjob.utias.utoronto.ca, Senior Member AIAA.
‡ Principal Research Engineer, Haimes@mit.edu, Member AIAA.

Within the context of aerodynamic optimization, the geometry control system is required to fulfil the following functions:

1) Given an initial geometry, determine a suitable set of parameters that represent the geometry with sufficient accuracy and serve as a set of design variables that provide the desired degree of flexibility for the optimization process.
2) Generate a new geometry in response to changes in the design variables. It should be possible to modify some or all of the design variables. It should also be possible to modify alternative design variables that are not necessarily the underlying parameters of the parameterization.

In order to facilitate grid regeneration, the following function is also desirable:

3) In response to the change in the geometry, provide a new surface grid, which retains the topology of the original surface grid and provides the basis for perturbing the volume grid.

We consider both a CAD-free and a CAD-based geometry control system. There are several trade-offs between the two approaches. We are developing the two systems for different purposes. The CAD-free system is intended for academic studies of relatively simple geometries. The CAD-based system is suitable for geometries that are more complex and for industrial use. Even in a practical design context, it may be worthwhile to perform preliminary studies, for example generating Pareto fronts and examining trade-offs, using the CAD-free system before switching to the CAD-based system for detailed design. Similarly, it may be preferable to optimize individual components using the CAD-free system before optimizing the complete configuration with the CAD-based system.

Geometry control systems for two-dimensional optimization have been limited to knowledge-based and free-form strategies[4]. Knowledge-based systems describe the airfoil in terms of physical coefficients such maximum thickness, camber, and leading edge radius. Free-form systems represent the geometry with linear combinations of basis functions such as B-splines[1,2,3]. Neither of these systems can be directly extended to three-dimensional geometries.

In three dimensions, CAD-free systems directly modify the discrete surface or often employ spline surfaces to parameterize the discrete surface. B-spline surfaces provide the required curvature control, the ability to perform local optimization, and the ability to apply geometric constraints[4]. The inherent difficulty with a CAD-free system is transferring the optimized geometry into the CAD environment[6] or into other systems. This challenge is offset with access to the entire source code, which permits easy customization and efficient grid perturbation.

An effective CAD-based system requires a CAD package based on parametric design and feature modeling. All of the operations in the geometry's history, and the parameters associated with the operations, are contained in its feature tree. Through the CAD package's Application Program Interface (API), these parameters form the CAD-based system's geometry definition, which can be altered to generate a new instance of the part. The API also provides the necessary geometric information required to perturb the volume grid. This permits the geometry to remain in its native CAD environment where it is available to all systems throughout the design and analysis cycle without loss of geometry information.

The major difficulties with a CAD-based system relate to the dependence on proprietary software. The interface and parameterization are inevitably dependent on the CAD system and licensing for parallel optimization procedures can become an issue[7]. The user must also be skilled enough to generate geometries with the parameters necessary to build the set of design variables at the start of the design and analysis cycle. In addition, grid perturbation becomes a greater challenge due to the possible topological changes in the parametric CAD environment after the design variables have been modified.

CAD-free and CAD-based systems must also provide a set of design variables for the optimization routine. Systems that modify the discrete surface directly may treat every grid point as a design variable[8] or apply displacement functions to the grid points. A common set of displacement functions incorporate bump functions. The bump functions allow the user to define the displacement at a number of positions and interpolate the displacement between these positions. An interesting combination of displacement functions are sine bumps combined with analytical functions for leading and trailing edge droop, leading edge bluntness, twist, dihedral and the planform shape[9]. Genetic optimization algorithms have also been implemented with displacement functions. One such system uses four variables to drive bump functions and one to define the twist at both the root and tip for a total of ten design variables[10]. Delta functions have also been defined by a combination of splines. One system uses Bezier curves to define the thickness distribution of defining airfoils and interpolates the thickness between these airfoils. It also uses Bezier surfaces to define the camber along the wing and B-spline curves to define the twist at each airfoil. In addition to the splines, it includes several variables to define the wing planform[11].

CAD-free systems that utilize spline surfaces to parameterize the surface grid have a control net as the foundation of their design variables. Gradient-based optimization routines with unstructured meshes have grouped the Bezier net into camber, thickness, and twist design variables on single and multi-element wings[12]. Non-uniform

rational B-splines (NURBS) have also been used to generate a constrained geometrical representation. This constrains the control net in the geometry creation stage and provides a reduced number of design variables from the free parameters[4].

CAD-based systems generate their design variables from the parameters built into the creation of the CAD geometry's parametric definition. Although the definition is dependent on the CAD package and how the geometry was generated, creating lofts from sketches that include splines is a common approach. Such a system based on Pro/ENGINEER and the Computational Analysis Programming Interface (CAPRI) has been implemented with Cartesian meshes on a fuselage, wing, canard, and tail geometry. Its design variables include the relative position of the volumes and coordinates of the defining splines[7].

B. CAPRI

CAPRI is an Application Programming Interface (API) whose purpose is to provide a seamless bridge between CAD systems and Computational Engineering Analysis (such as CFD). This middleware is designed so that the differences found in the internals of each geometry kernel and CAD system are hidden from the programmer/user. The API is uniform, without special cases, and vendor neutral. The result is that when a CAPRI application is written it can run for all of the supported systems without modification.

The data model in CAPRI consists of a combined geometry and topology (solid BRep) view of the part as well as a discrete triangulated, watertight perspective. This associated dual view can provide a complete, easy to use, access point into the CAD data. Hands-off meshing is possible by starting with the tessellation and enhancing through either physical or parameter space manipulation, using point "snap" and *[u, v]* surface evaluation routines provided by CAPRI.

CAPRI contains the following major API components:

1) The reader component is the basic level of support allowing for loading, managing, tagging, and querying symbolic topology and geometry.
2) The Solid Creation and Solid Boolean operation component supports the making of simple solids and intersecting, subtracting and fusing any solid loaded into or created from within CAPRI. This supports multi-disciplinary analysis and optimization where each discipline shares some geometric components (such as fluid/structure interaction).
3) The Master Model component allows for parametric modification, control of certain geometric shapes, and defeaturing of parts. A parametric CAD system is required to properly build the part and to provide this level of geometric modification and control.

CAPRI supports the geometry kernels Parasolid and OpenCASCADE, and the CAD system CatiaV4 with the first two components listed above. The CAPRI ports to Pro/ENGINEER, I-DEAS, CatiaV5, and UniGraphics provide all three API components.

The CAD-based system described in this paper is the first application of CAPRI using CatiaV5 in a design setting.

C. Objectives

The primary objective of this study is to develop a CAD-free and a CAD-based geometry control system for three-dimensional aerodynamic shape optimization with the following three functions:

1) Parameterize an initial geometry into a set of hierarchical design variables that allow for varying levels of freedom and differing design methodologies.
2) Modify the geometry based on changes in the design variables.
3) Perturb the surface grid to conform to the modified geometry.

The two geometry control systems provide practical tools for aerodynamic optimization. They also provide a basis for comparing CAD-free and CAD-based systems and understanding additional issues that need to be addressed in order to develop reliable optimization tools.

II. Methodology

A. CAD-Free Methodology

The CAD-free system includes a surface extractor, a surface fitter, a control net perturbator, and a surface generator. Figure 1 illustrates their relation to the complete optimization cycle starting with an initial volume grid that is passed to both the surface extractor and grid perturbator. The solid and dashed lines denote initialization and optimization cycles respectively.

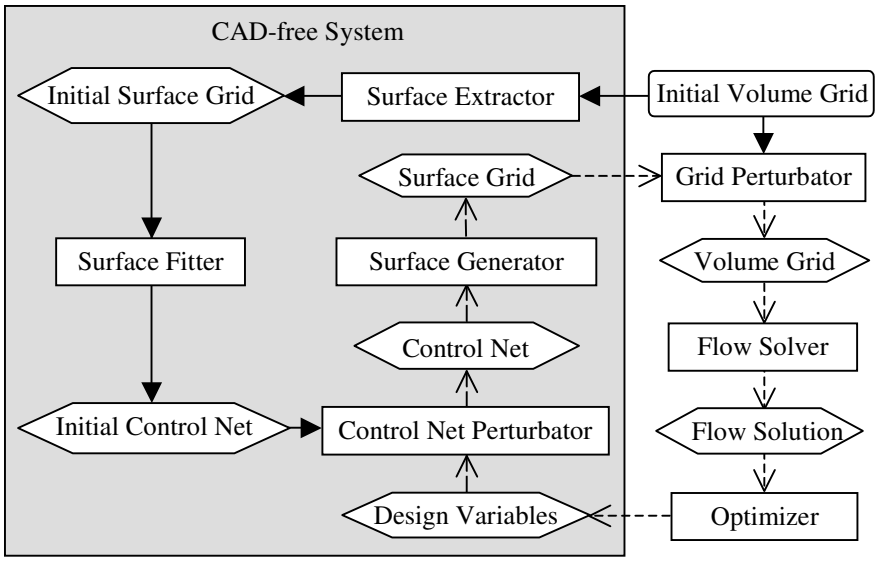

Figure 1. CAD-Free flow chart.

The surface extractor reads the initial volume grid along with a surface description file. The surface description file includes the point numbers representing the origins of each surface patch, and the number of points in the *u* and *v* directions for each patch. This information facilitates surface extraction and defines the B-spline surface patches.

The surface fitter fits B-spline tensor product patches to the initial surface grid and passes the initial control net to the control net perturbator. The control net perturbator modifies the initial control net based on the design variables that are driven by the optimizer. With the perturbed control net, the surface generator then regenerates the surface grid for the grid perturbator.

Once the grid perturbator has the modified surface grid, it perturbs the volume grid, which is passed to the flow solver. The flow solver can then produce the flow solution, which the optimizer analyses to determine the next set of design variables. With the new set of design variables, the optimization cycle can continue until the user defined objective function is minimized.

B. CAD-Based Methodology

The CAD-based system includes a CAD reader and writer, a grid applicator and extractor and a boundary modifier, which are the five CAPRI components, as well as an object modifier and a two-dimensional grid perturbator. Figure 2 illustrates their relation to the complete optimization cycle starting with the initial CAD geometry. The solid and dashed lines denote the initialization and optimization cycles respectively.

To begin the optimization cycle the CAD reader extracts the parametric geometry description from the initial CAD geometry and builds the initial geometry object. This object contains all of the geometric information required to describe the components of the geometry to be optimized. The object modifier makes the changes to the initial geometry object dictated by the optimizer via the design variables and generates the modified geometry object. The CAD writer then modifies the parameters in the initial CAD geometry based on the modified geometry object to generate a new instance of the CAD geometry.

At the same time, a grid generator, such as ICEM, creates the volume grid from the initial CAD geometry. The grid applicator can then use the volume grid and the initial CAD geometry to generate the static, dynamic, and interface surface grid objects. The surface grid objects contain all of the information required to describe the surface grid on the CAD geometry. This includes the volume, section, and partition numbers and normalized *u*-*v* coordinates of each point in the surface grid.

The static surface grid object defines all of the surface grid points that are not affected by a volume interface or hole. The dynamic surface grid object defines the points that must be altered due to a volume interface or hole and the interface surface grid object defines the boundary of the hole or volume interface. The boundary modifier alters the *u* and *v* coordinates of the points contained in the interface surface grid object due to changes in the geometry. These modified interface points can then be used to drive the two-dimensional grid perturbator, which generates the modified dynamic surface grid object.

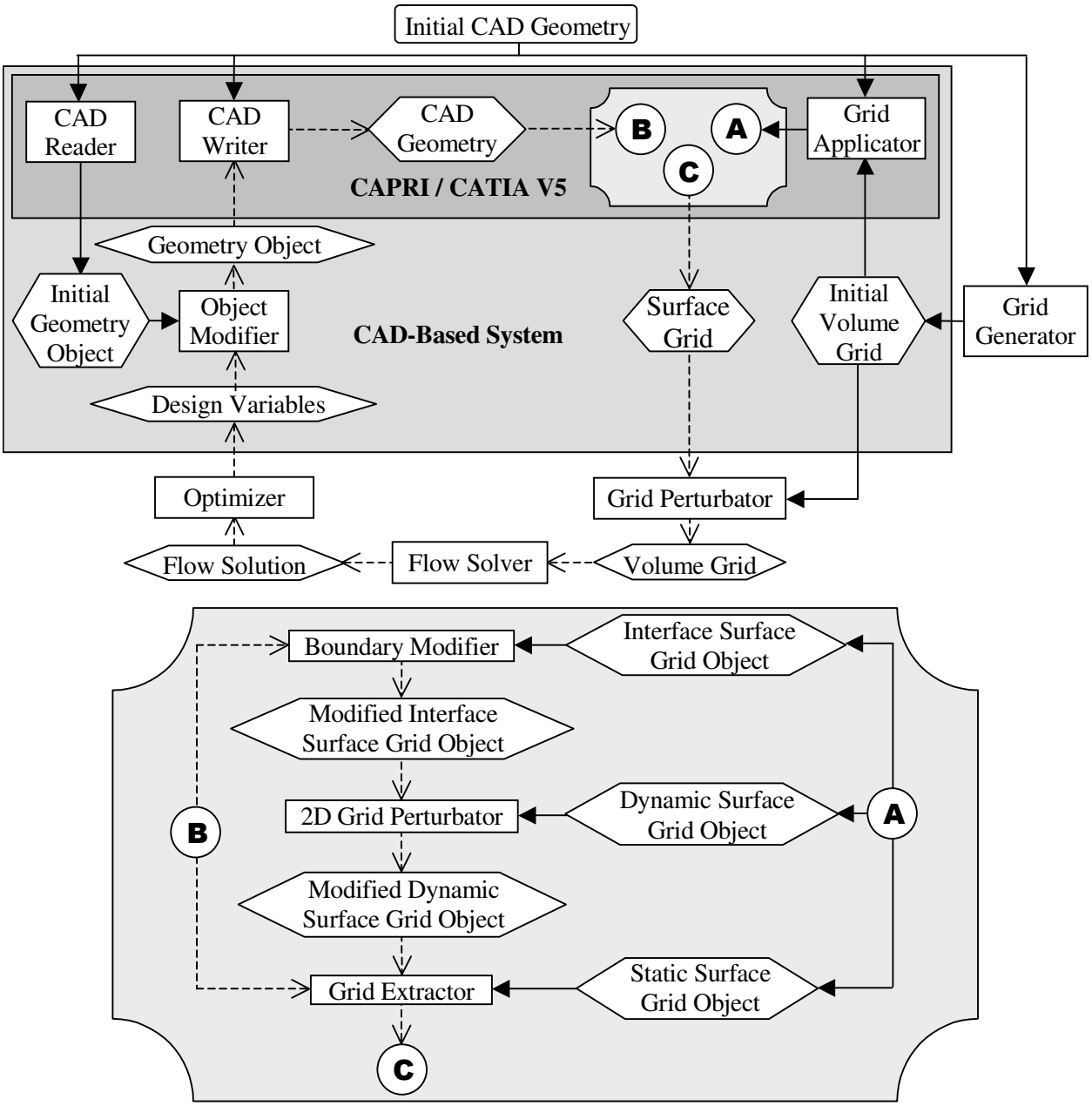

Figure 2. CAD-Based flow chart.

With the static and modified dynamic surface grid objects, the grid extractor can then interrogate the modified CAD geometry to generate the modified surface grid. This modified surface grid is passed to the grid perturbator to complete the optimization cycle.

When the objective function is minimized, the final optimized geometry is simply the last instance of the CAD geometry. This ensures that all of the information contained in the original geometry is transferred to the optimized CAD geometry. In addition, this system makes the modified CAD geometry available to other systems throughout the optimization cycle without any loss of information.

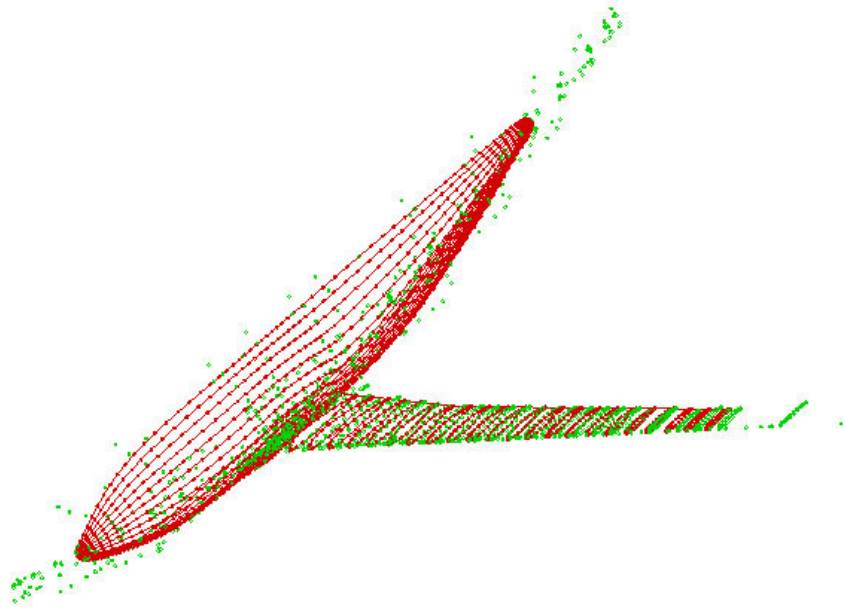

Figure 3. Sample B-spline parameterization.

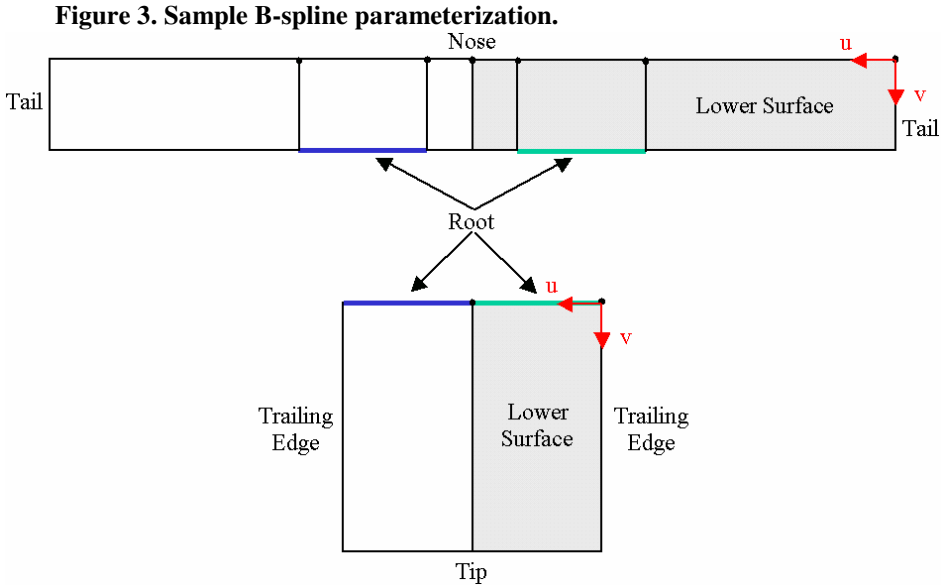

Figure 4. Sample B-spline patch configuration.

III. Parameterization

A. CAD-Free Parameterization

The CAD-free system parameterizes the geometry from the initial surface grid with B-spline tensor patches. The user must supply the origin and number of points in the *u* and *v* directions of each patch before the surface grid is extracted from the volume grid. With this information, the surface grid is divided into patches and a B-spline patch with an open uniform knot vector is generated. The order of the patches may be changed but it must not change across patch boundaries. The number and position of the control points on neighbouring patch boundaries also have forced coincidence. Along with the open knot vector and equivalent order, this guarantees a watertight geometry.

Figure 3 illustrates the B-spline patches applied to the DLR-F6 geometry without under-wing nacelles and pylons from the Second Drag Prediction Workshop[13]. The green and red points represent the control net and surface grid, respectively, and the control net is comprised of 800 control points.

Figure 4 illustrates the eight patches that were required to parameterize the DLR-F6 geometry when disassembled. A single patch was applied to both the upper and lower surface of the wing but three patches were

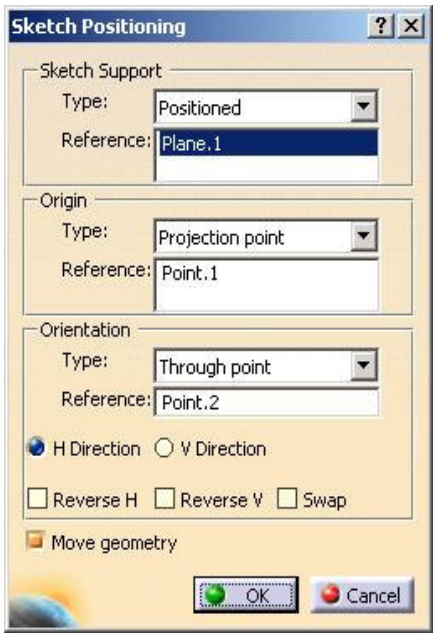

Figure 5. Sketch support positioning.

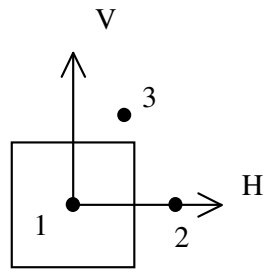

Figure 6. Support plane.

required for the both the upper and lower surface of the fuselage. The three patches were required to ensure a closed volume at the root.

B. CAD-Based Parameterization

The CAD-based system incorporates the CATIA V5 geometry kernel and leaves the geometry in the CATIA V5 environment. This is accomplished with the use of the Computational Analysis Programming Interface (CAPRI)[14,15].

The parameterization assumes that the components of the CAD geometry to be optimized are collections of lofted bodies. The lofts are generated from sketches containing a closed profile. The profile may contain any combination of wireframe objects such as lines, polylines and arcs, but only the spline sections may be modified. In addition, to modify the position and orientation of the sketch's support plane, the support type must be positioned, the origin must be a projected point, and the orientation must be set through a point. Figure 5 illustrates the required sketch support.

In addition to the two points mentioned above, one more point is required to define the sketch's support plane and orientation of the *v* axis. Figure 6 illustrates the relation between a support plane's axis system and its three defining points. Note that the third point is not required to lie on the *v* axis.

To facilitate modification of the support plane's orientation or position, a parameter for each coordinate of its three defining points must be added to the loft and a relation must be made between the coordinate and the parameter.

C. CAD Geometry Generation

The manual creation of geometries with the required parameters for CAD-based optimization is time consuming and error prone even for an experienced CAD operator. Therefore a geometry generation routine has been developed that reduces the time required to create multi-bodied geometries from hours to minutes. This also allows for the addition of interpolated sketches to increase the number of design variables.

The geometry generation routine utilizes a macro-parametric approach[9] to generate and parameterize the original geometry. First, a setting file is read that contains filenames of the 2D defining profiles and their scale, orientation and position. From the position and rotation, the coordinates of the three points that define the support plane are calculated. From the original coordinates and the associated scale, the coordinate profiles contained in each sketch are also calculated. With the defining sections, the interpolated profiles are generated. If the number of points in adjacent sections differs, a B-spline curve is used to normalize the profiles. Once all of the profiles are created, a CATIA V5 script is generated that creates the desired geometry with all of the required parameters.

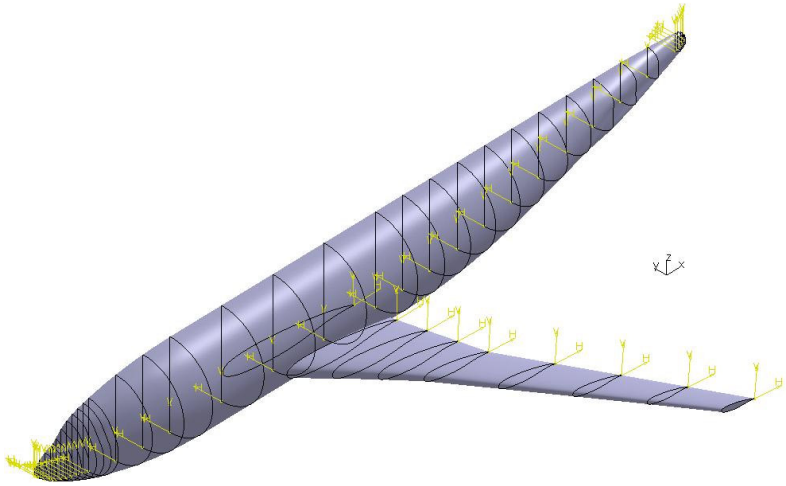

Figure 7. Sample CAD-based GCS geometry.

The resulting geometries can be the building blocks of more complex geometries. The routine can be called several times to generate multiple bodies. These bodies can then be added to the same CATIA part to create geometries that are more complex. For instance, the routine can create a wing, a fuselage, a pylon, and a nacelle, which can be joined. Internal and external details can also be added in CATIA to the base geometry, such as spars, antennae, and detailed part information.

This approach was used to generate the DLR-F6 geometry shown in Fig. 7. In this example, only four profiles were required to generate the wing but nine were used. The extra profiles are redundant in the initial geometry but they increase the number of design variables in the spanwise direction. This provides the optimizer with more control along the span.

IV. Design Variables

A. CAD-Free Design Variables

The CAD-free design variables have three layers for varying degrees of control. These are the B-spline control net, driving nodes, and global variables.

The control net provides excellent geometry control, including the ability to locally modify the surface[4].

The driving nodes constrain the control net by generating control point displacements. The displacements to be added to the control points between the driving nodes are sinusoidal functions of the nearest driving nodes in the *u* and *v* directions. This smooth displacement distribution has been previously implemented as sine bumps in aerodynamic shape optimization[9].

```
Planform Global Variables
 10   Leading edge sweep
 10   Trailing edge sweep
 10   Dihedral
 10   Angle of attack
-10   Twist
 10   Chord
 10   Span
 10
 10   Wing displacement
-10
Airfoil Driving Nodes
  2   Number of driving airfoils
  0
 50   Position of the driving airfoils
  1   Number of driving nodes
 20   Position of the driving nodes
  0
  0   Free axes
  1
 10
 10   Node deltas
Fuselage Driving Nodes
  2   Number of driving cross-sections
 25
 36   Position of the driving cross-sections
  1   Number of driving nodes
100   Position of driving nodes
  0
  0   Free axes
  1
-20
  0   Node deltas
Fuselage Global Variables
  1   Number of driving cross-sections
 56   Position of driving cross-sections
-40   Radius deltas
```

Listing 1. Sample B-spline design variables.

The global variables organize the control net displacements into variables that are applied to an entire volume. This provides the designer with an intuitive set of design variables and leads to meaningful designs[16]. For instance, a designer may wish to evaluate the effects of sweep, dihedral, twist, aspect ratio, taper, wing position or fuselage cross sectional area on an existing geometry. The cost of manufacturing non-linear geometries, especially in sections containing control surfaces, also increases the desire to optimize the geometry with these linear geometric functions.

Listing 1 illustrates a sample set of design variables for a wing-fuselage geometry. Sweep, dihedral, angle of attack, and twist are measured in degrees from the original value. Chord, span, and radius are measured in percent

offset from the original values. Wing and fuselage displacements are percentages of the original root chord and fuselage length, respectively, and positions are normalized with zero and one hundred at the start and end of the associated geometry.

B. CAD-Based Design Variables

Unlike the CAD-free design variables where the actual parameterization can be optimized, the CAD-based design variables do not directly include its parameterization. To ensure a valid geometry, a two-layered set of design variables modifies the underlying parameterization. All rotations and translations of the sketches are accomplished by modifying the points that define the support plane. To scale or modify the shape of the sketch profiles, the coordinates of the splines contained in each sketch are modified.

The first layer of design variables contains nodal variables analogous to the B-spline's driving nodes. With the CAD-based design variables, these displacements are added to the two dimensional coordinates of the splines contained in each sketch.

```
Global Variables
 -10
  10   Taper
  10
  10   Skew
  10
  10   Rotation
 -10   Twist
  10
 -10   Scale
 -10
 100
-100   Displacement
 200
Nodal Variables
   1   Number of driving sections
 100   Position of driving sections
   3   Number of driving nodes
  20
  50   Position of the driving nodes
  75
   0
   1   Free Axes
 200
   0   Driving Node displacements
 200
```

Listing 2. Sample CAD-based design variables.

The second layer contains global variables that are very similar to the CAD-free global variables. The similarity is most evident when the CAD-based design variables are applied to a wing with the u axis in the chord direction from the trailing edge to the leading edge. In this case, tapering the wing in the u and v directions modifies the leading edge sweep and the thickness distribution along the span. Skewing the wing in the u and v directions alters the trailing edge sweep and dihedral. Rotation and twist control the wing's angle of attack and twist distribution. Scaling the wing in the u, v and w directions adjusts the mean chord, thickness and span respectively, and finally displacing the wing in the u, v and w directions moves the wing in the u, v and w directions of the first airfoil sketch.

The main difference between CAD-free and CAD-based design variables is that the CAD-based variables are generic and may be applied to any CAD loft created with the appropriate parameters. In contrast, both the nodal and global CAD-free design variables are divided into wing and fuselage types. This specialization resulted from complete access to the CAD-free source code and geometric parameterization, which makes tailoring of the design variables fast and easy. The CAD-based variables are based on the more generic and structured CAD-based parameterization, which ensures that all lofted bodies can be optimized.

Listing 2 includes a sample design variable file for a simple wing. All variables are measured in the same manner as the CAD-free design variables.

V. Surface Grid Perturbation

The generation of a body-fitted-structured or unstructured volume grid can be computationally expensive and often requires user intervention to ensure smoothness. It is highly desirable to avoid this complete grid generation, especially during gradient-based optimization in a complex design space. This leads to grid perturbation and Cartesian grid generation routines.

Cartesian grid generation routines have proven to be robust and essentially automated for multi-bodied and multi-volume geometries[7]. Due to the complete grid regeneration, however, a smooth transition between grids after even small changes in the design variables may introduce noise into the optimization cycle. This noise may influence the efficiency of gradient-based optimization routines[7].

Grid perturbation routines help maintain a smooth transition between grids by tracking the surface grid through geometry modifications. This displaced surface grid then perturbs the volume grid in a smooth manner. Due to large

changes in the geometry's topology or relative position between bodies, it may be necessary to completely regenerate the surface grid. In many practical design problems, however, the geometric constraints prohibit major topological changes.

The two systems developed here assume that a grid perturbation routine is desired. Therefore, the position of each point in the surface grid relative to the initial geometry must be saved and used to determine its three-dimensional coordinates on the modified geometry. Special care must also be taken to ensure that the points near face boundaries such as the wing root are properly modified and the volume remains watertight.

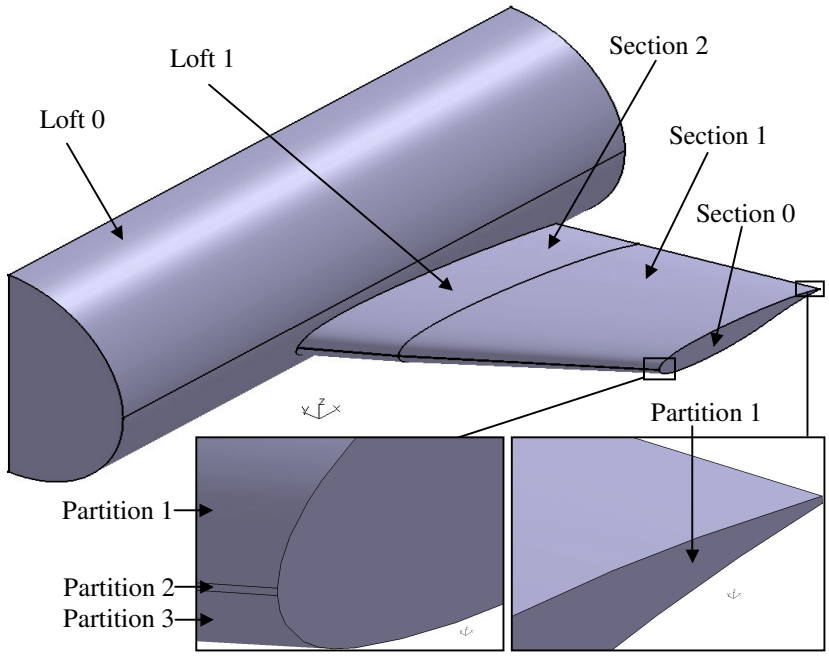

Figure 8. CAD-based internal geometry representation.

A. CAD-Free Surface Grid Perturbation

Since the CAD-free system utilizes B-spline patches that are fitted to the original surface grid, perturbing the surface grid is extremely fast and robust. After the initial fit, the original u, v coordinates and the patch number of each point is saved. This information is used after each geometry modification to regenerate the surface grid. This ensures a smooth transition through the geometry modification. In addition, all points on a patch boundary shall remain on the shared boundary, thereby guaranteeing a watertight volume.

Even though this system guarantees a successful and smooth grid perturbation, it may be desirable to regenerate the surface grid after gross geometry modification to maintain grid quality.

B. CAD-Based Surface Grid Perturbation

Perturbing the surface grid with the CAD-based system is more complex than with the CAD-free system. This results from the CATIA V5 parametric modeling. After each geometric modification and the associated change in the parameterization, CATIA completely regenerates the geometry. This may cause changes to the number of faces used to enclose each volume and the range of u and v on each face. In the CAD-free system, the number of faces is fixed and the u and v coordinates always range from zero to a fixed maximum. To overcome this dynamic geometric representation, an internal representation of the volume and each point is generated.

This representation exploits the fact that all of the components of the geometry to be modified are lofts. Figure 8 illustrates a simple geometry consisting of a fuselage and a wing. The black lines in the figure represent face boundaries. This representation organizes the faces into lofts, sections, and partitions. The lofts obviously represent the lofts of the CAD parameterization. The sections represent the faces between the sketches that define the CAD loft. If only one end of the loft is uncovered, such as the wing, the section numbering begins with the uncovered end. If both or neither of the ends are uncovered, such as the fuselage, the numbering starts at the end nearest to the first sketch that defines the CAD loft. The maximum number of sections is the number of sketches in the associated loft plus one. In Fig. 8, the fuselage has two sketches and three sections. The wing has three sketches but only three sections because one of the ends is inside the fuselage.

The partitions are areas of a section where the u coordinate is continuous across face boundaries. Partitions are required when CATIA V5 splits faces on a section in the u direction to maintain a closed volume. This often occurs when a spline on a sketch is modified in such a manner to create a region of high curvature. In Fig. 8, the leading and trailing edges at the wing tip are magnified to view the four partitions on section one of loft one.

Using the CAD-based internal representation, the CAD-based system gives each point in the original surface grid a loft, section, and partition number. The u and v coordinates of each point on the CAD face are then normalized with respect to the boundaries of its partition.

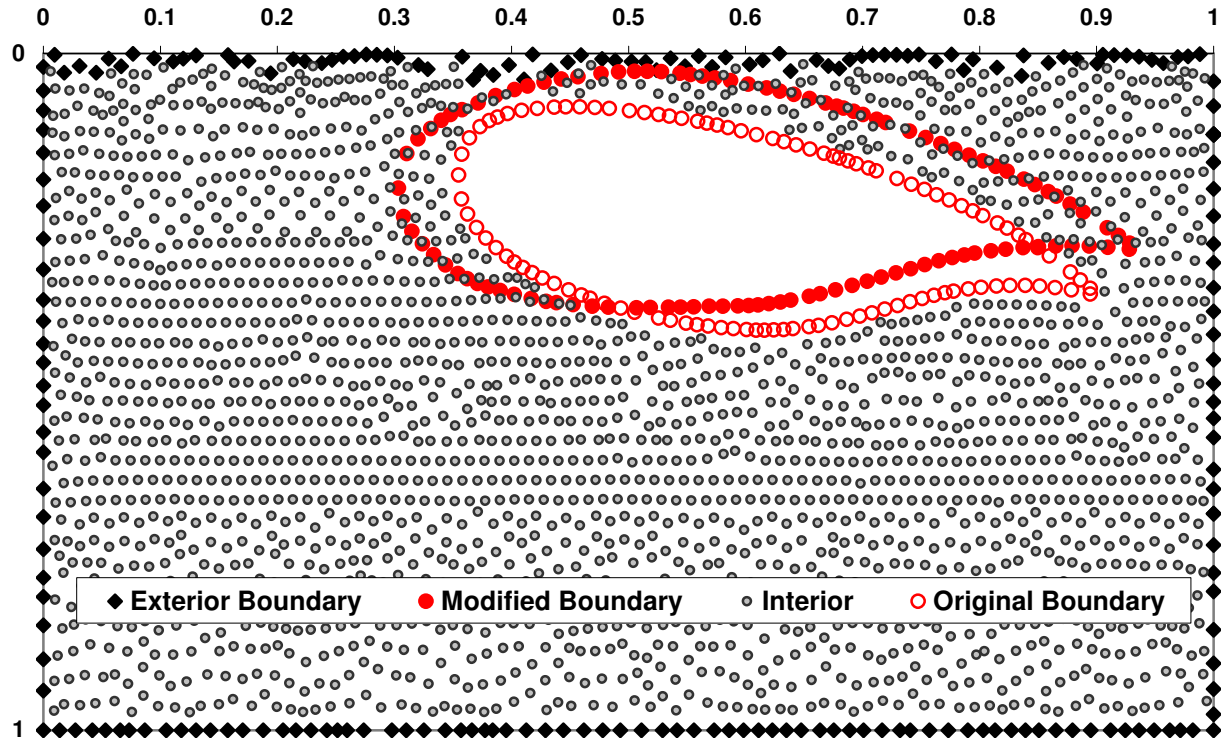

Figure 9. A parametric surface grid at a volume interface.

After the geometry modification, the geometry is interrogated and each face is assigned a loft, section, and partition number. The *u* and *v* coordinates of each point are then de-normalized based on the new partition boundaries. With the loft, section, and partition numbers and the de-normalized coordinates, the CAD-based system determines the associated CAD face and extracts the new three-dimensional coordinates. These three-dimensional coordinates represent the perturbed surface grid.

The CAD-based system encounters another difficulty on partitions that include a hole or volume junction. The problem arises because the face parameterization does not change as the loop that defines the hole or volume junction is modified. Figure 9 illustrates a fuselage partition that includes a wing fuselage junction in parametric space. The black diamonds represent the partition boundary, the grey squares represent the interior points of the partition, the hollow circles represent the original wing-fuselage junction, and the solid circles represent the junction position after the geometry has been modified.

As Fig. 9 illustrates, after the geometry has been modified, the loop that defines the junction moves in parametric space but the fuselage face is unaffected. Therefore, the *u* and *v* coordinates that define the fuselage surface grid points near the junction may now exist inside the wing root, or may exist further from the root. This prevents a watertight surface grid from being generated. To overcome this difficulty, a two-dimensional grid perturbator is applied to the *u* and *v* coordinates of the points on the partitions that include one or more loops with a negative sense.

VI. Results

A. CAD-Free Results

The CAD-free system was applied to the CATIA V4 DLR-F6 geometry without under-wing nacelles and pylons supplied by the Second Drag Prediction Workshop[13]. CATIA V5 converted the V4 geometry into a V5 CATPart file. ICEM CFD 4.2 then read the V5 geometry, generated the initial structured volume grid with its Hexa Mesher, and exported it in the Plot3D format with a super domain above and below the geometry.

The CAD-free system approximated the original surface grid with the four fourth-order B-spline patches illustrated in Fig. 4. Twenty control points were used on each of these four patches in both the *u* and *v* directions. With these settings the surface fitter approximated the original surface grid with a mean, median and mode error of $5.05\text{x}10^{-5}$, $1.01\text{x}10^{-5}$ and 0.0000000 when the dimensions were normalized to the wing's half span.

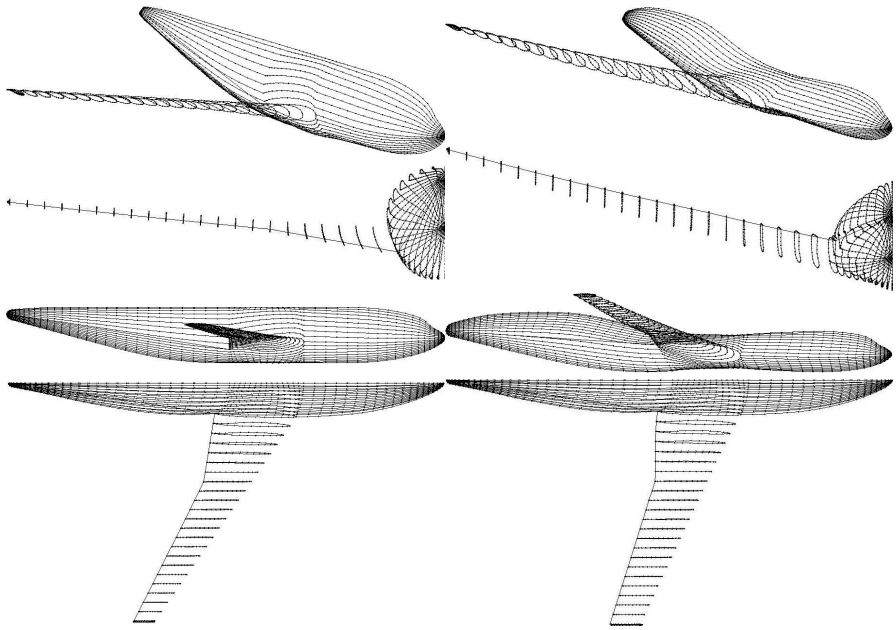

Figure 10. CAD-free geometry and surface grid perturbation.

With the CAD-free system initialized, the design variables illustrated in Listing 1 were used to perturb the geometry and its surface grid. Figure 10 illustrates this operation. Note how the fuselage morphs to accommodate changes in the orientation and position of the wing's root. When the design variables force a patch's boundary to be altered, the effects are smoothed into the neighbouring patch with a sinusoidal weighing. This ensures a closed volume and avoids discontinuities across patch boundaries.

B. CAD-Based Results

The CAD-based system is applied to the same geometry from the Second Drag Prediction Workshop[13] as used by the CAD-Free system. The CATIA V4 geometry is not parametric nor feature based. It is simply a collection of geometric primitives, such as points, lines, splines, and NURBS surfaces. The CAD-based system requires a parametric and feature based solid model. To generate the necessary parameterization, CATIA V5 interrogates the initial geometry to generate the sections that define the geometry. The "Advanced Meshing Tools" of CATIA V5 outputs a discretized version of the defining sections that are read by the Geometry Generator. The Geometry Generator then creates a new version of the geometry that has the parameterization required to generate all of the desired design variables.

This process approximates the non-parametric geometry with a parametric and feature based model. Since the original non-parametric geometry does not contain the method by which it was generated, it is impossible to guarantee that the parametric geometry is a perfect approximation. If the degree to which the parametric model approximates the initial geometry is a great concern, the method by which the initial geometry was generated should be imitated. For instance, if lofting or sweeping a spline created the initial surfaces, then this process must be replicated.

From the properly parameterized CATIA V5 geometry, ICEM CFD 4.2 generates the initial volume grid with its unstructured Tetra Mesher and exports it in the Star-CD format. To test the surface-grid-generation routines, the design variables were set to zero, and the surface grid of 3109 points was regenerated. The regenerated surface grid has a mean, median, mode, and maximum error of $1.59\text{x}10^{-5}$, $9.65\text{x}10^{-6}$, $1\text{x}10^{-6}$, and $3.05\text{x}10^{-4}$ normalized to the wing's half span.

With the CAD-based system initialized, the design variables in Listing 2 are used to perturb the geometry and its surface grid as illustrated in Fig. 11.

VII. Conclusions and Recommendations

Two geometry control systems have been developed for three-dimensional aerodynamic shape optimization. They both utilize a hierarchical set of design variables to drive modifications in the original geometry and the surface grid.

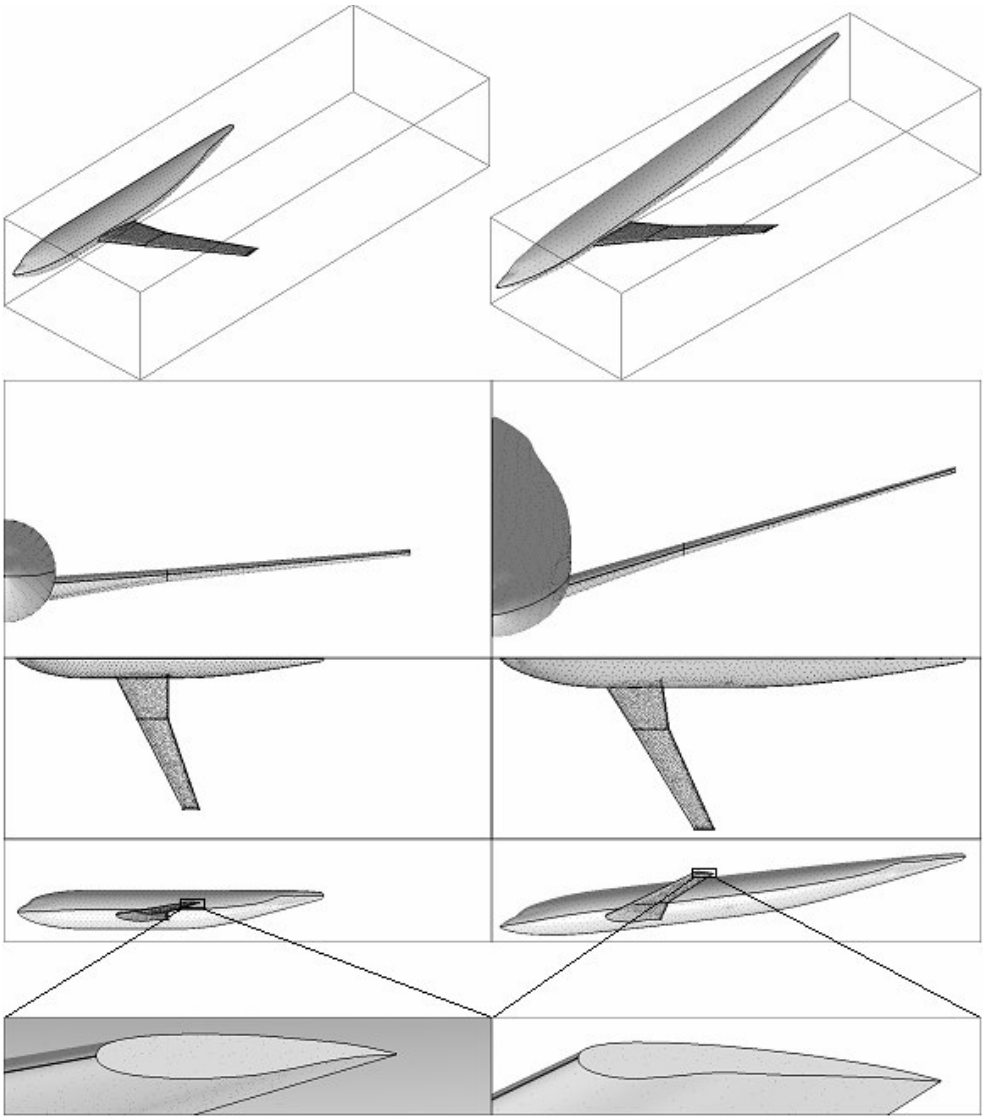

Figure 11. CAD-Based geometry and surface grid perturbation.

The CAD-free geometry control system parameterizes the initial surface grid with B-spline patches. To provide the optimization routine with varying degrees of geometric control, the CAD-free system implements three layers of design variables with the third layer divided into planform and fuselage design variables. The nodal and global variables can be used to drive the B-spline control net or the net may be modified directly. CAD-free surface grid modification is accomplished by interrogating the B-spline patches at fixed u and v coordinates.

The CAD-based system's parameterization is built on the parametric and feature-based solid modeling of the CATIA V5 geometry kernel. This allows the CATIA V5 geometry definition to be interrogated and modified with the use of CAPRI. Unlike the CAD-free design variables, where the actual parameterization can be optimized, the CAD-based design variables do not directly include its parameterization. To ensure a valid geometry, a two-layered set of design variables, nodal and global variables, modifies the underlying parameterization. The nodal and global CAD-based design variables are generic and can be applied to any volume. Perturbing the surface grid with the CAD-based system is more complex than with the CAD-free system due to the parametric modeling in CATIA V5. To overcome this dynamic geometric representation, an internal representation of the volume is generated and interrogated after the geometry modification to modify the surface grid.

The CAD-free system can parameterize any structured surface grid where along any line of constant v, u increases continuously from $u_{\min}$ to $u_{\max}$. The CAD-free design variables are configured to smoothly perturb

structured surface grids that define wing or wing-fuselage geometries. Since the developer can modify the complete CAD-free parameterization and source code, it can be tailored to nearly any geometry.

The CAD-based system can parameterize and modify complex geometries with multiple connected and disconnected volumes. This includes more than one complete aircraft configuration. Its surface grid perturbation routine is robust when applied to single-bodied volumes and multi-bodied volumes with fixed intersections. The surface grid can also be perturbed about dynamic volume intersections with the use of a two-dimensional grid perturbation routine. Since the geometry never leaves its native CAD environment, no information is lost during the optimization cycle.

To increase the flexibility of the CAD-free system, it should be extended to unstructured grids and geometries that are more diverse. If the initial geometry is used to generate both a structured and an unstructured grid, the structured grid could be used to parameterize the geometry and the unstructured grid could used to generate the u-v coordinates. Minimizing the error between the initial x, y, and z coordinates and the x, y and z coordinates of the point on the B-spline surface created from the structured grid could generate the desired u and v coordinates. This minimization would only occur during the initialization of the CAD-free system and it is very similar to the Grid Applicator of the CAD-based system. The CAD-free code was written with the intention to extend it to geometries that are more complex. The major change to the code would be related to extending the design variables to other topologies.

Currently two situations can prevent the CAD-based system from perturbing the surface grid at a dynamic volume interface. The first situation occurs when a section completely disappears inside the intersecting volume or a section originally hidden inside the intersecting volume is exposed. The CAD-based system could be designed to catch the creation or destruction of sections, and redistribute the grid points among the sections. If the geometry is generated with the section boundaries sufficiently far from the volume boundary, the destruction or construction of a section represents a significant geometric change. After this change, a complete regeneration of the surface grid may be more desirable than a surface grid perturbation to ensure grid quality.

The second situation occurs when a dynamic volume interface crosses a partition boundary. The two-dimensional grid perturbation routine does not allow for a discontinuity in the u or v coordinate that exists at a partition boundary. To overcome this problem the CAD-based system could create a super-partition across the intersection. This problem could also be prevented by ensuring that the initially volume interfaces are not located near a partition boundary.

It should be noted that these difficulties are confined to surface grid perturbation at volume interfaces and arise from two factors: (1) During the (re)generation of geometry, the CAD system performs the required Boolean operations on the underlining solids and freely splits up surfaces to define the resulting volume. Therefore, the surface topology at the volume interface may change between iterations. (2) The Newton-Krylov (gradient-based) optimization requires sensitivities of the geometry to changes in the parameters. Because it is impossible to differentiate through the CAD system, this information must be derived by differencing of surface points from various instances of the model. Hence the problem: how does one track points if the topology changes? This was addressed by constructing partitions in the CAD-Based system. It was also addressed in Alonso *et al*[18] by imprinting a consistent topology over the one supplied via the CAD system where all resultant faces were logically quadrilaterals. Points were then placed via a TFI-like scheme within each patch where CAPRI was used to "snap" the points to the underlying surface. This approach is not a general solution to the problem in that the CAD model (in this case Pro/ENGINEER was used) was marked up with the imprinted topology, approximately doubling the complexity of the component.

The general solution would be a superset of the scheme that Dannenhoffer and Haimes[19] are developing. In this work, CAD faces are combined where the separation is due to construction artifacts or sliver entities in a similar manner as the creation of the CAD-Free partitions. The procedure of splitting these "quilts" will be required in order to be able to achieve a specific topology. With these operations and a prescription of the desired outcome, a consistent topology is realizable from CAD regardless of the BRep supplied (assuming that there is no actual topological change).

Finally, if the optimization scheme does not require sensitivities (i.e. simulated annealing or other non-gradient method) then neither artificial nor fundamental changes in topology would be an issue. The CAD-based system presented here would be robust and require no intervention, assuming the rest of the analysis can handle a changing surface representation in a hands-off manner.

VIII. References

[1]Marian Nemec, David W. Zingg, "A Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," AIAA Journal, Vol. 40, No. 6, June 2002, pp. 1146-1154.

[2]Marian Nemec, David W. Zingg, "Optimization of High-Lift Configurations Using a Newton-Krylov Algorithm," AIAA Paper 2003-3957, 16th AIAA Computational Fluid Dynamics Conf., Orlando, June 2003.

[3]Marian Nemec, David W. Zingg, Thomas H. Pulliam, "MultiPoint and Multi-Objective Aerodynamic Shape Optimization", AIAA Journal, Vol. 42, No. 6, June 2004.

[4]Luc Huyse, Sharon L. Padula, R. Michael Lewis, Wu Li, "Probabilistic Approach to Free-Form Airfoil Shape Optimization under Uncertainty", AIAA Journal, Vol. 40, No. 9, Sept., 2002.

[5]Anas Bentamy, Jean-Yves Trepanier, Francois Guibault, "Wing Shape Optimization Using a Constrained NURBS Surface Geometrical Representation", ICAS Congress, 2002.

[6]Guk-Heon Choi, Duhwan Mun, Soonhung Han, "Exchange of CAD Part Models Based on the Macro-Parametric Approach", International Journal of CAD/CAM, vol. 2, 2002.

[7]Marian Nemec, Michael J. Aftosmis, Thomas H. Pulliam, "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method", AIAA 2004-0113, 42nd AIAA Aerospace Sciences Meeting, January 5-8, 2004, Reno NV.

[8]R. Lohner, O. Soto, C. Yang, "An Adjoint-Based Design Methodology For CFD Optimization Problems", AIAA-2003-0299, 41st AIAA Aerospace Sciences Meeting & Exhibit, Reno, Nevada, Jan., 2003.

[9]Susan E. Cliff, Scott D. Thomas, Timothy J. Baker, Antony Jameson, "Aerodynamic Shape Optimization using Unstructured Grid Methods", AIAA 2002-5550, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 4-6 September 2002.

[10]Terry L. Holst, Thomas H. Pulliam, "Aerodynamic Shape Optimization Using a Real-Number-Encoded Genetic Algorithm", AIAA 2001-2473, 19th AIAA Applied Aerodynamics Conference, Anaheim, California, 11-14 June 2001.

[11]Daisuke Sasaki, Shigeru Obayashi, Keisuke Sawada, Ryutaro Himeno, "Multiobjective Aerodynamic Optimization of Supersonic Wings Using Navier-Stokes Equations", European Congress on Computational Methods in Applied Sciences and Engineering, Barcelona September 11-14, 2000.

[12]Eric J. Nielson, W. Kyle Anderson, "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes", AIAA Journal, Vo. 40, No. 6, June, 2002.

[13]Neal J. Pfeiffer, "Reflections on the Second Drag Prediction Workshop", AIAA 2004-0557, 42nd AIAA Aerospace Sciences Meeting, January 5-8, 2004, Reno NV.

[14]Robert Haimes, "CAPRI: Computational Analysis Programming Interface, A Solid Modeling Based Infra-structure for Engineering Analysis and Design", CAPRI user's guide, MIT, Revision 1.00, Dec. 18, 2000.

[15]Robert Haimes, "Computational Analysis Programming Interface, Uniform Vendor-neutral CAD Access through an API", CAD-CFD Workshop, University of California, April 1999.

[16]Kasidit Leoviriyakit, Antony Jameson, "Aerodynamic Shape Optimization of Wings including Planform Variables", AIAA 2003-0210, 41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan., 2003.

[17]Antony Jameson, Luigi Martinelli, Bob Haimes, "Aerodynamic Shape Optimization of Complete Aircraft Configurations using Unstructured Grids", 42nd AIAA Aerospace Sciences Meeting and Exhibit, 5-8 January 2004, Reno Nevada.

[18]Juan Alonso, Joaquim Martins, James Reuther, Robert Haimes and Curran Crawford, "High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model", Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA 2003-3429. Orlando FL, June 2003.

[19]John Dannenhoffer and Robert Haimes, "Using Quilts and Chains to Improve Structured and Unstructured Surface Grids", AIAA Paper 2004-0610, 42nd AIAA Aerospace Sciences Meeting and Exhibit, 5-8 January 2004, Reno Nevada.