UNIFIED GEOMETRY ACCESS FOR ANALYSIS AND DESIGN

Robert Haimes[1] Curran Crawford [2]

[1] *Aerospace Computational Design Laboratory, Massachusetts Institute of Technology, U.S.A*
haimes@mit.edu
[2] *Aerospace Computational Design Laboratory, Massachusetts Institute of Technology, U.S.A*
ccrawfor@mit.edu

ABSTRACT

This paper presents a comprehensive approach for CAD based geometry handling in support of single and multidisciplinary analysis and design. Unlike previous schemes, the model presented here allows for *hands-off* automated meshing, a requirement for design studies. Multidisciplinary analysis is handled through solid modeling constructs, using the geometry as a transfer media. For design applications, the engineer specifies the design space directly by the parameters of the CAD model. Key defining values in the model specified during part synthesis are later exposed to carry out design studies and optimization. Proper layout and definition of the CAD model facilitates design changes and use across all stages of design. A turbine blade model suitable for detailed aero/structural/thermal analysis is used to illustrate these concepts.

Keywords: CAD interfaces, Parametric build, Feature tree, Master-Model, Analysis, Design, Optimization

1. INTRODUCTION

There are a number of projects [1, 2, 3] and some commercial software [4, 5, 6] that facilitate building complex, multidisciplinary analysis, optimization and design systems. These software products either provide closed systems or can take existing applications and through encapsulation allow the building of complex custom applications. This is primarily a *top-down* view where each module requires a number of inputs and produces some output. The infrastructure's function is to control the flow of data between modules. None of these systems can alleviate a bottleneck in a discipline, and at the simplest level these software systems just minimize the need for input and output files. These efforts primarily deal with discrete design variables, objectives and constraint functions, rather than complete geometric descriptions of the system.

When analyzing (or designing/optimizing) some physical object that will ultimately be manufactured, it is common practice to create a geometric definition in a CAD system. Erroneously, the CAD model is often only a final repository of the design details. For most disciplines (all except for Structural Analysis), the bulk of time (and human intervention) is expended in transforming this data into a mesh that is suitable for the physics to be analyzed. In some cases, such as for Computational Fluid Dynamics (CFD), this time period can be weeks to months when first setting up a new complex part. Clearly, one cannot hope to do a parametric study, no less design optimization, under these conditions. The problems associated with grid generation stem from a number of sources, with the most obvious being the use of inadequate file 'standards' for transmitting the geometry.

The commonly used IGES file format contains data that is defined as disjoint and unconnected surfaces and curves; that is, it only contains geometry with no notion of topology. 3D meshing software ultimately requires a closed "water tight" model. Much effort is therefore needed to take the geometric data, trim the curves and surfaces, and then deduce the topology. A common side effect of this process is the creation of many "sliver" surfaces to close the model. This process is particularly onerous for all modern CAD systems where solid modeling is fully supported. Before translation, the part was a proper closed solid with defined topology, the CAD system having been responsible for verifying the necessary conditions; in the translation to IGES this important characteristic is needlessly lost.

The STEP file format supports topology as well as geometry. This is therefore the preferable file type to use for the transmittal of CAD neutral data. Surprisingly, this format is seldom used in practice. This may be due to the fact that constructing a STEP reader is complex and requires a complete solid modeling geometry kernel to deal with the data. Also, transferring data via STEP is not without its own set of problems. Each CAD system uses a different mathematical formulation to represent the same types of surfaces and also have different tolerances for closure. After reading a solid part, one may find the model is now open, again requiring some form of patching. An additional problem with data transmitted via IGES and STEP formats are that the writers rarely are in strict adherence to the "standard". Standards can only be as good as the extent to which they are followed.

These problems do not exist for some native CAD/Analysis interfaces. Ansys, MARC, and Patran (as well as other widely used commercial codes) couple directly to CAD systems. These analysis codes are designed to be activated from within the graphical user interface of the CAD package. The part's geometry and topology are queried directly from the CAD's geometry kernel using an Application Programming Interface (API). Curve and surface queries through the originating CAD system insure that points placed on these entities match the part being meshed. These direct interfaces are inherently limiting, in that a module must be pre-existing for the analysis code of interest (i.e. what CFD codes are run from within the CAD environment). Furthermore, the full functionality of the analysis package may not be available from within the CAD system, limiting the user's options for exploiting the software's full capabilities.

Even if all of the analysis codes in an integrated suite have no geometry-handling issues the following question needs to be asked: is each application seeing the same representation of the design? Some codes with direct CAD interfaces still translate the geometry into a simplified or uniform definition before using the data to generate a mesh. Many organizations also maintain multiple models of the same part within different divisions, introducing unnecessary overhead.

2. A UNIFORM DIRECT INTERFACE

The grid generation techniques used in a discipline such as CFD are more varied and complex than the (relatively coarse) tetrahedral meshes used for Structural Analysis. Due to the economics (the enormous amount of work) of coupling to each CAD system, this approach is prohibitive for other "smaller" disciplines. A direct vendor neutral API would allow an analysis builder access to the CAD data without programming directly for each system. Examples of this approach include OMG CAD Services [7], CADScript [8], and CGM [9]. CAPRI [10] (Computational Analysis PRogramming Interface) also provides a solution to the CAD dependency issue. Coupling to any supported CAD package is both unified and simplified by using the CAPRI definition of geometry (with topology) and its API to access the geometry and topological data.

CAPRI's CAD-vendor neutral API is more than just an interface to CAD data; it is specifically designed for the construction of complete analysis suites. A 'Geometry Centric' approach allows access to the CAD part from within all sub-modules (grid generators, solvers and post-processors), facilitating such tasks as node enrichment by solvers and designation of mesh faces as boundaries (for the solver and the visualization system). CAPRI supports only manifold solids at its base level, eliminating problems associated with manually closing surfaces outside of the underlying CAD kernel. Multidisciplinary coupling algorithms can use the actual geometry as the medium to interpolate data from differing grids.

One clear advantage to this approach is that the geometry never needs to be translated and hence remains simpler and closed. The other major advantage is that writing and maintaining the grid generator (coupled to the CAD system) can be done *once* through the API; all of the major CAD vendors are then automatically supported.

2.1 CAD Representation of Geometry

CAD systems have a tolerance that determines the meaning of "closure" for solids. This means that the Nodes that bound an Edge are probably not on the underlying curve; Edges that bound a Face (through the Loops) do not necessarily sit on the supporting surface. All that is required is that the bounding objects be within a specified tolerance of the higher dimensioned entity. Therefore, for any precision higher than the tolerance, gaps and overlaps may exist in the geometry definition. This tolerance is generally much larger than those associated with double precision floating-point arithmetic (e.g. the default relative tolerance for Pro/ENGINEER is only 10^{-2}).

In order to deal with the gap and overlaps, most CAD-based applications must "fix" the geometry. This usually entails translating the geometric definition to another (simpler) representation where the bounding entities fall closer to, or on the object. This type of translation has a variety of side effects, including:

- Inconsistency: Not querying the same geometry. Since the geometry has changed, the representation is different than in the CAD system.

- Complexity: At times additional Faces are required to close the model. There is no way to predict how many of these "sliver faces" may need to be introduced; moreover, slivers can cause significant problems for grid generators.

- Automatic: There are always situations that cannot be healed in a *hands-off* manner. The requirement of user intervention is problematic for any fully automated process such as design optimization.

CAPRI's perspective is that the geometry in the CAD system is *truth* and should not be modified (though CAPRI may modify the topology). Therefore fixing the CAD's model is no longer part of the analysis procedure.

2.2 An Associative Triangulation

Early in the design and implementation of CAPRI, it became obvious that providing an API only giving the programmer access to the geometry and topology of a solid part was insufficient. The burden of deciphering the CAD data and attempting to generate a discrete representation of the surfaces required for mesh generation was too great. Fortunately, many grid generation systems (used in CFD and other disciplines) can use STL (*Stereo Lithography*) files as input. Combining a discretized view of the solid part as well as it's geometry and topology can provide a complete, and easier to use, access point into the CAD data. A tessellation of the object that contains not only the mesh coordinates and supporting triangle indices but other data, such as the underlying CAD surface parameters (for each point), as well as the connectivity of the triangles, assists in traversing through and dissecting the CAD representation of a part. This is a fundamental difference between CAPRI and the other Direct Interface implementations.

An important aspect of CAPRI is that it provides CAD vendor neutral access to all of the data obtained from the models that is to be passed back to the application. The triangulation generated by CAPRI is guaranteed to be "watertight", regardless of the CAD kernel in use. Some CAD system geometry kernels can provide data of this quality (i.e., UniGraphics, Parasolid, CATIA and ComputerVision). Other CAD systems can provide the data, but it is not of sufficiently high quality to use. (For example, Pro/Engineer requires one to buy Pro/MESH to get a closed triangulation.) Finally, SDRC's Open I-DEAS API does not provide access to a triangulation at all. The fact that not all CAD systems provide such a tessellation has forced the development of a surface triangulator within CAPRI for CAD solid parts that *does* meet all of the quality requirements.

It should be noted that CAPRI's tessellations are not intended as the starting point for computational analysis (though they could be used in some cases). CAPRI sees only geometry, and it cannot anticipate the smoothness, resolution or other requirements of the downstream application(s). The triangulations approximate the geometry only; some processing of the tessellation is expected in order to refine the triangulation to a state suitable for the physical problem being investigated. The triangulation can be enhanced through either physical or parameter space manipulation, using point "snap" and (u, v) surface evaluations routines provided by the CAPRI API [11]. The triangulation technique used within CAPRI displays the following characteristics [12]:

- Robust. It is imperative that the scheme work for all possible topologies and provide a tessellation that can be used.

- Correct. The triangulation is of no use if it is not true to the CAD model. The tessellation must be logically correct; i.e. provide a valid triangulation in the parameter space (u, v) of the individual surface. It must also be geometrically correct; i.e. depict a surface triangulation that truly approximates the geometry. This involves ensuring all facets have a consistent orientation with no creases or abrupt changes in triangle normals. Correctness in both physical and parameter space allows CAPRI based application enhancement schemes to operate in either or both.

- Adjustable. To minimize the post-processing of CAPRI's tessellation for a specific discipline or analysis, some a priori adjustment of the resultant quality is available. It must be noted however, that any criteria may not be met (especially near the bounds of a CAD object) due to issues of closure and solid model accuracy. This goal may conflict with the more important characteristic of being watertight and having a smooth surface representation.

- No geometric translation. To truly facilitate hands-off grid generation, anything that requires user intervention must be avoided. All data maintained within CAPRI is consistent with the CAD's solid model representation. An alternate or translated representation is not used, because then the result will be something different than resides within CAD.

- Watertight. Triangulated CAD solids are closed and conformal; having this characteristic allows for meshing without "fixing" geometry. For the tessellation of a solid object, this means that all Edge (trimming) curves terminate at consistent coordinates of the bounding Nodes and a single discretization for Edge curves be used on both surfaces sharing the common Edge. Each triangle side in the tessellation is shared by exactly two triangles, and the star of each vertex is surrounded and bounded by a single closed loop of sides. The triangulation is everywhere locally manifold. In a manifold triangulation, there are no voids, cracks or overlaps of any triangles that make up the solid.

3. MULTIDISCIPLINARY ANALYSIS

The classic example of multidisciplinary analysis is the modeling of fluids/structures interactions. In this case, the domain of interest is clearly demarcated between regions containing the fluid and the rest of the volume representing the structural components. The interaction effects are usually handled at the bounds of the shared domain (the interface), represented by some surface (or set of surfaces) found in the solid geometry (usually of the structural component). The interaction can be simulated by adjusting the boundary conditions at the interface surface(s) from the computed results of the other discipline. Most commonly, pressure values on the structure are obtained from the CFD calculation, causing regions of the structure to deflect. This movement then effects the fluid simulation, which generates a new pressure field, et cetera. In most cases, the CFD simulation can take the structural deflections and *morph* the existing mesh so that the fluid domain need not be regenerated. This still requires careful handling of the following issues:

3.1 Geometry

It is important that each discipline see the same interface. There are two reasons for this: (1) so that each analysis is examining results from the same part; (2) so that the interpolation task can be done with some degree of accuracy.

CAPRI supports multidisciplinary analysis tasks by supplying API calls that implement solid Boolean operators. Since the intersection, subtraction and union operators are performed by the CAD system's geometry kernel, the end results contain the same geometry fragments. For example, if the solid is of an airplane, subtracting it from a larger box will produce the fluids domain. The surfaces that make up the bounds of the aircraft will be identical in both the original and the new solid (though the normals will be opposite).

3.2 Interpolation

CAPRI facilitates multidisciplinary coupling by providing interpolation routines. These routines work on sets of Faces (the topological equivalent of surfaces) called Boundaries. Information associated with the points in the Boundary can be scalar, vector or general state-vector. This data can be interpolated to the points of the mating Boundary found in the other Volume. CAPRI provides a number of API functions that help manage the Boundary information so that the interpolation can be a single call.

4. WRITING GEOMETRY

At beginning of the CAPRI project there was always the notion that design functionality would be supported. At the time, it was thought that CAPRI would support the direct construction of 3D solid geometry in order to allow for the modification of said geometry. As the readers were being implemented, it became obvious that this would not be possible. Each CAD system deals with the low- level geometry construction in very different manner. There was not a common vendor neutral perspective on direct construction. In fact, only those systems based on geometry kernels (and allowing the use of the kernel) could perform construction. Therefore, only if one programmed in Parasolid, ACIS or OpenCASCADE could this kind of construction be performed.

As it turns out, this limitation was fortunate; another type of construction was required that could be driven by an API. Most modern CAD systems support the Master-Model concept of representing an object. A Master-Model describes the sequence of topological operations to build the geometry of a solid model. At a basic level, it is an ordered list of extrude, revolve, merge, subtract and intersection operations. CAD systems support more meaningful abstractions, such as blends, fillets, drilled holes and bosses. When the CAD model is regenerated, the operation list is interpreted by the CAD system to sequentially build the geometry of the part. This gives the operator the ability to construct a family of parts (or assemblies) by building a single instance. Many of the operations used in the construction can be controlled by parameters that may be adjustable. By changing these values, a new member of the family can be built by simply following the prescription outlined in the Master-Model definition.

The recipe may be simple, like a serial collection of primitive operations, but can also be complex, where operations are performed on previously or temporarily constructed geometry. The representation of this construction in most CAD systems is the form of a tree, usually referred to as the "Feature Tree". By supporting this method of construction, a direct API can provide both simple and powerful access to the CAD system. This approach is clearly outside the static view traditionally held of geometry. That is, this kind of access and control is not possible from any type of file transfer.

Within CAPRI, this tree is presented to the programmer in the form of "branches". Each of these entities has an index to identify where in the tree the reference is made. All indices are relative (that is they can occur anywhere in the tree – the assignment is usually given during initial parsing of the CAD internal structures). There is a special branch always given the index zero, the root of the tree. Therefore, the entire tree may be traversed starting at the root and moving toward the end of each branch. The branches terminate at leaves (branches that do not contain any children). To aid in traversing the tree toward the root the parent branch is always available. Unlike simple binary trees, a branch in CAPRI's Feature Tree may contain zero or more children.

Currently, the structure of tree itself cannot be edited from within CAPRI (though this may change at some future release). However, some branches may be marked "suppressible" – these features may be turned off, in a sense removing that branch (and any children of the branch) from the regeneration. This is powerful in that it allows for defeaturing the model, so that it may be made appropriate for the type of analysis at hand. For example: if fasteners are too small for a fluid flow calculation, they may be easily suppressed (if the Master-Model was constructed with this in mind). After part regeneration the resultant geometry would be simplified and the details associated with the fasteners would not be expressed.

Parameters are those components of the Master-Model that contain values (and should not be confused with the geometric parameterization). CAPRI exposes all of the adjustable (non-driven) parameters found in the model. This is a separate list from the Feature Tree, but references back to the associated branch features where the values are used or defined. Parameters may be single or multi-valued and can be Booleans, integers, floating-points or strings.

This CAD perspective on parametric building of parts and assemblies is fine for driving the part using simple parameters but is problematic for shape design. For example, simple parameters may be used to define the plan-form of an aircraft, but are difficult to use to define the airfoil shape of the wing and tail components. The designer would need to expose the curve/surface definition at a very fine and detailed level (i.e. knot points as the parameters) to allow for the exact specification of shapes. CAPRI avoids placing this burden on the CAD designer by exposing certain curves as multi-valued "parameters". These curves are obtained from independent sketched features in the model that later are used in solid generation as the basis for rotation, extrusion, blending and/or lofting. The curves can be modified, and when regenerated, the new part expresses the changed shape(s). This functionality is critical for shape design in general and specifically aerodynamic shape design.

5. AN UPSTREAM VIEW

The traditional design process (in many fields) starts from a conception stage where no actual geometry may be specified, to a final design where the part is fully realized down to the finest details. In a multidisciplinary design setting, one discipline may set some "parameters" before passing its information along to the next. Only when there is the requirement for more detailed analysis requiring geometric properties will the design be *fleshed out* and placed into a CAD system in a solid representation. It should now be clear that if the design process changes from this traditional situation to one where the designer predefines the part's intent and possible expression (through a Master-Model definition) the following becomes feasible:

- Consistency. Each phase in the design process uses the same suite (or a subset of the suite) of parameters. Any parameter value change that produces differing geometry can be viewed by another stage in the process without writing and reading the geometry in files. The CAD part, regenerated with a particular set of parameter values and Feature Tree suppression statuses, uniquely describes the geometry.

- Data Repository. The CAD system and Product Data Management (PDM) software can be used to track and maintain the design. Also, because the design is in the CAD system from the beginning, issues of manufacturability can be easily addressed early on and unrealistic expressions kept out of the design space.

- Use of defeaturing to go from preliminary to final design. If the Master-Model is built in a manner that reflects the design process, then traversing the stages in the process is just a matter of adjusting the Feature Tree. During preliminary design where the resultant geometry may be simple (or nonexistent) most of the branches of the tree are suppressed. As the design approaches the final intent, more and more of the details of the part are expressed by unsuppressing the branches. This will also require setting various parameters as their effects become active.

- Use of defeaturing for various disciplines. Suppressing branches of the Feature Tree can also be used to match the fidelity of the geometry to the analysis being performed. For example, if CFD is being used and the meshing scheme cannot handle fillets, then the fillets can be suppressed. This is a much simpler and more rigorous approach than trying to modify the fully expressed part after the fact (and it can be done automatically).
- Parameter studies and design optimization. The designer has specified the parameters (hopefully) in a meaningful manner. This means that parameter studies can become as simple as setting a new value, having the CAD system regenerate the geometry and then analyzing the new instance. A complete design space can be mapped out from the complete set (or subset) of the parameters. This means that the process of automated design can be tracked and some insight gained into the design by visually tracing the selection of parameter values.

In order for the proposed approach to be successful, the designer must understand the nuances of the CAD package in use to robustly define features that will persist across the family of parts. The parts must be put together with care, to ensure that the appropriate dimensions in the model are driven by meaningful parameters. Also, features must be created in such as way as to allow later suppression and modifications to the CAD model as the design matures. For example, a simple box with filleted edges should be created as an extrude feature of a rectangle, followed by another feature defining the fillets on the edges. The alternative approach is to extrude a rectangle with filleted corners. The latter representation will make it impossible to later suppress the fillets for lower fidelity analysis, and will also tend to break the CAD model should the fillets later be deleted from the design.

The CAD model should be constructed so that Master-Model effectively captures the decomposed intent of the design, starting with the most basic definition through to the finest manufacturing details. To achieve an even higher level of modularity and to more fully capture the design *intent*, complex parts should be modeled inside an assembly, especially for aerodynamically constrained applications. This approach will be illustrated in the next Section.

6. A TURBINE BLADE EXAMPLE

A turbomachinery blade model is manipulated to illustrate ideas expressed in the previous Section. The complete representation of the blade can be seen in Figure 1.a The source model was constructed using Pro/ENGINEER.

It should be noted that all of the figures in this Section were generated from CAPRI's triangulation directly and without any massaging. Also, all modifications to the Master-Model were done using CAPRI functionality and not Pro/ENGINEER's Graphical User Interface.

This solid part model contains 134 parameters. 12 of these parameters are the curves gleaned from four sketches in the model. The sketches are located at the root, tip and two mid-span locations on the blade, and each contains curves used to define the suction and pressure surfaces, in addition to a third curve defining the camber-line (metal turning angles at the leading and trailing edges connected by a tangent spline). 13 are floating point and integer named parameters, which include things like trailing-edge thickness, wall thickness, and number of blades in the row, etc. The remaining parameters are what Pro/ENGINEER calls *dimensions*. These are unnamed values that are required by the sketcher to fully determine sections (CAPRI also exposes these values when not driven and treats them the same as Pro/ENGINEER *parameters*).

The Feature Tree for the model contains 233 nodes. This may seem excessive until one realizes that the turbine blade is internally cooled. Figure 1.b displays the same part as seen in Figure 1.a except that the exterior and interior Faces that bound the pressure surface of the blade have been removed, exposing the interior of the blade. The triangulation of the interior of the suction surface wall has been highlighted to better display the structure. The internal flow enters through a hole that can be seen on the left had side of the root (under the hub surface). The fluid travels up the left (leading edge), being tripped by the ridges that can be seen (and do not completely block the passage). The cooling air continues to follow the serpentine until it is back down at the root in the center of the blade. The flow leaves the slot in the trailing edge as it continues to be mixed by the pins blocking the exit.

Figure 1.c depicts just the internal void of the blade displayed as a solid. Here the entire internal flow path can clearly be seen.

Figure 2 shows the turbine blade at various levels of feature suppression. Figure 2.a displays the turbine blade with all of the internals removed (i.e. this is a solid blade). In this view, the only visible difference is the absence of a hole at the root. It should be noted that the solid seen in Figure 1.c was generated in CAPRI by performing a solid Boolean subtraction of the full part as seen Figure 1.a from this instance (Figure 2.a).

The fillets that merge the aerodynamic portion of the blade to the hub and tip casements have been removed

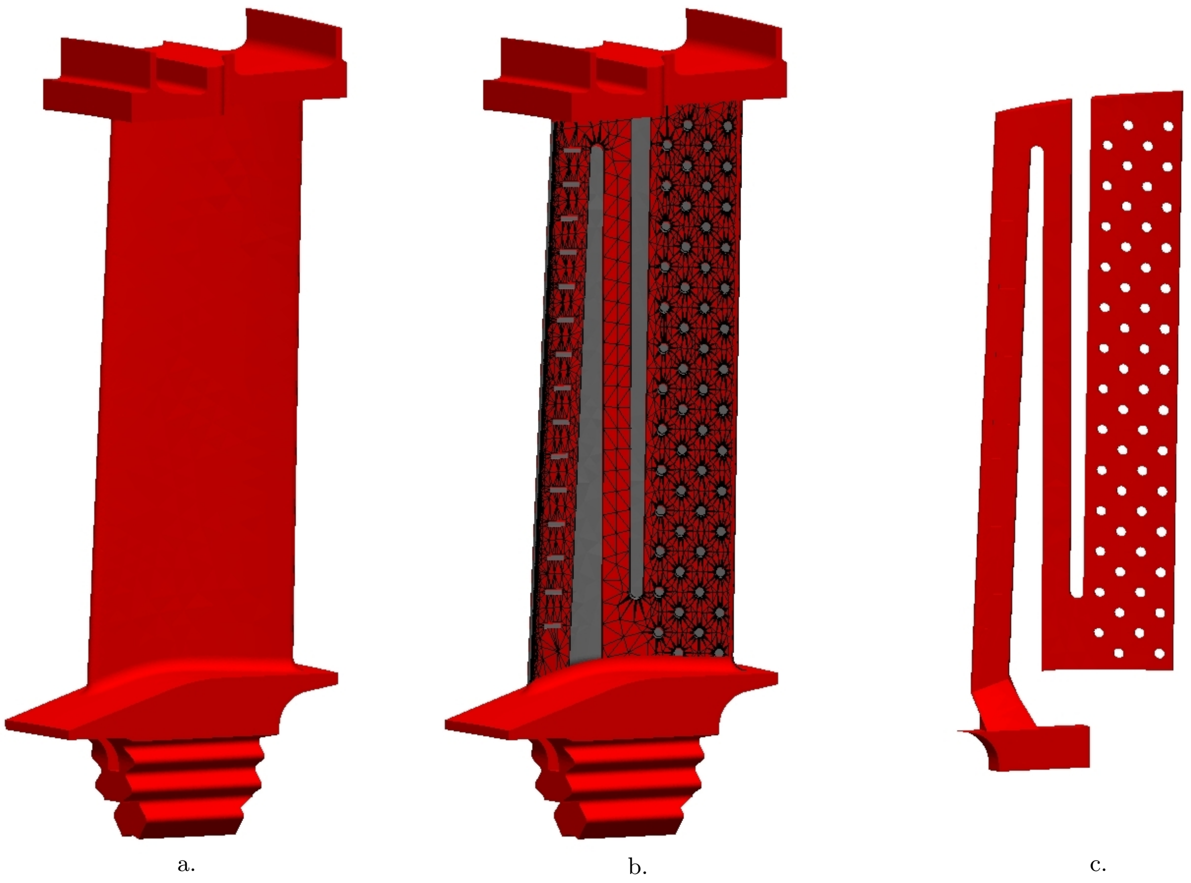

a. b. c.

Figure 1: a) The internally cooled turbine blade, b) pressure surface removed to display internal features, c) the internal flow path as a solid

in Figure 2.b. Due to the view angle this difference can only be seen down at the root/blade juncture. Figure 2.c depicts the blade in its simplest solid form; all extraneous details of the hub and tip have been removed. This geometry could be used for preliminary and/or aerodynamic design.

When performing grid generation for CFD, one needs a solid representing the flow regime. A fluid volume suitable for analysis of the turbine blade using periodic boundary conditions is shown in Figure 3.a. This step illustrates the utility of using assemblies during part construction. The solid model of the blade is contained as the second part of an assembly. Analogous to parts, assemblies are also described by Master-Models. In this case, the first element in the assembly Master-Model is a specialized type of part, referred to as a skeleton in Pro/E parlance. The distinction is that it contains no solid geometry, only datums, which can include non-manifold surfaces.

The skeleton Master-Model contains two revolved surfaces, one for each of the inner and outer walls of the turbine stage annulus. The solid blade part then references these two surfaces to construct the root and tip shrouds on the blade, adding material to go from Figure 2.c to Figure 2.b. A third part was created in the assembly, again referencing the annulus surfaces to create a complete manifold solid over the entire circumference of the annulus, both upstream and downstream of the blade. The sketches of the metal turning angles in the blade part are then referenced to create the offset sides of the periodic wedge seen in Figure 3.a, by slicing the complete annulus. The dimensions of the wedge are determined by a parameter in the blade Master-Model specifying the number of blades in the row.

Modularizing the CAD model in the above manner effectively separates the design intent. The flow annulus design becomes an independent operation, while remaining a driving factor in the blade design. In this way, consistency is also maintained between the parts. The references used in each part are created from the published features of the other parts, affording further control over the model composition and avoiding non-robust topological constructs. For example, the

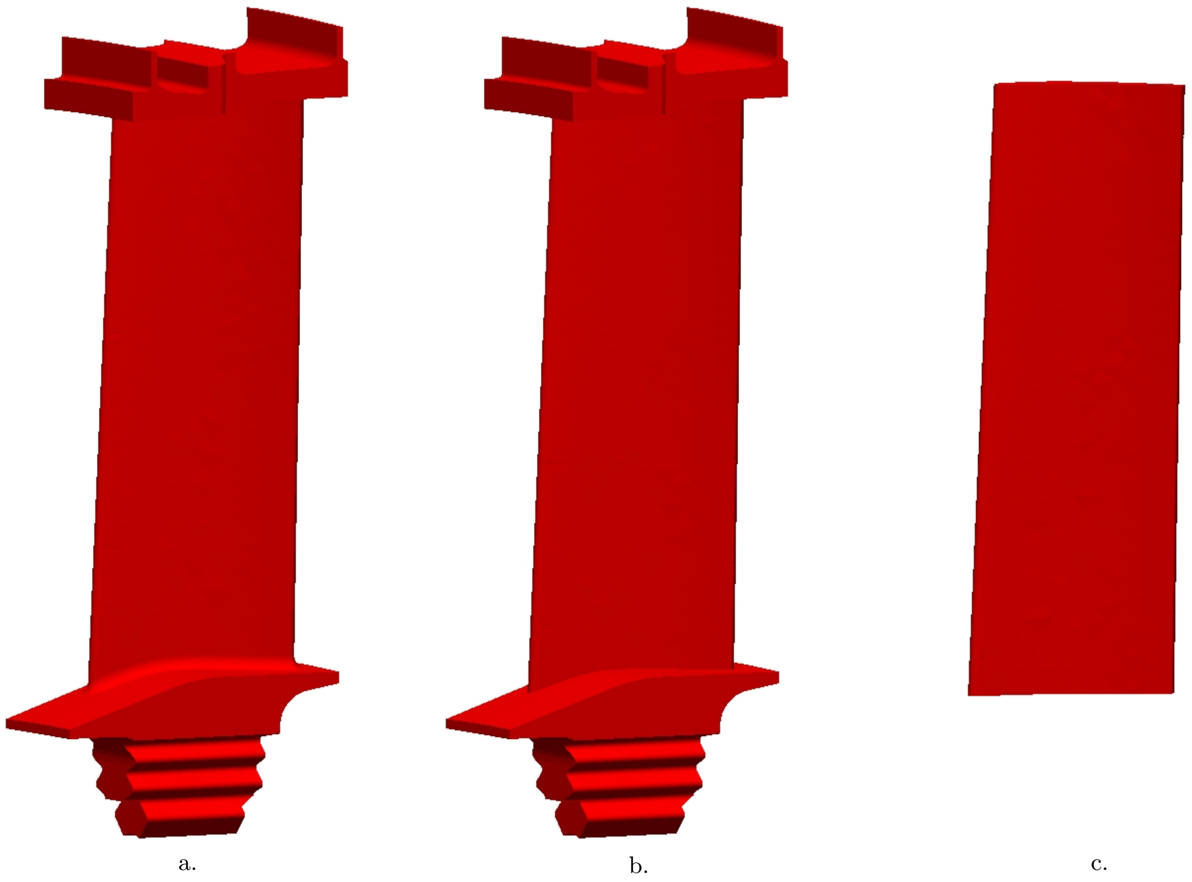


Figure 2: The defeatured turbine blade – a) no interior, b) no fillets c) only the aerodynamic blade

rotation axis of the rotor is made a common reference point between each part. As the design matures, the upstream and downstream rows of blades and stators will be able to reference the same annulus definition, again maintaining coherence between the CAD models.

In order to produce the complete CFD domain, as seen in Figure 3.b, the blade seen in Figure 2.c is subtracted from the blank passage of Figure 3.a. Figure 3.b is displayed with the periodic surface closest to the viewer stripped away to show the blade cut-out.

If it was desirable to compute on the complete flow regime (both internal and external), a solid Boolean union of the objects seen in Figure 3.b and Figure 1.c could be used. The result can be seen in Figure 4.a. This could also have been performed by subtracting the complete turbine blade (Figure 1.a) from the passage seen in Figure 3.a, but the inflow region (seen below the passage) would be absent, because the lower surface of the wedge did not extend below the hub surface definition.

Figure 4.b displays the effect of changing a parameter value. In this case the number of blades in the blade row was reduced by half. The result is that the casing treatments grew to accommodate the requirement of completing the circumference.

7. DISCUSSION

The approach articulated in this paper, with its precise control over the geometry, is quite powerful (as displayed in the previous Section). One can now have the CAD system central in an automated design optimization loop, without the encumbrance of user interaction. This approach does not replace the *top-down* approach but augments it well, producing a system that also has a consistent *bottom-up* geometry methodology.

It should be reiterated that when using a direct interface there is CAD vendor independence. The analysis modules are isolated from the details of the underlying back-end system. Also when the modifications to the model are made through the API, by the analysis application, there is no need to step back out to interactively use the CAD system (hence interrupting the

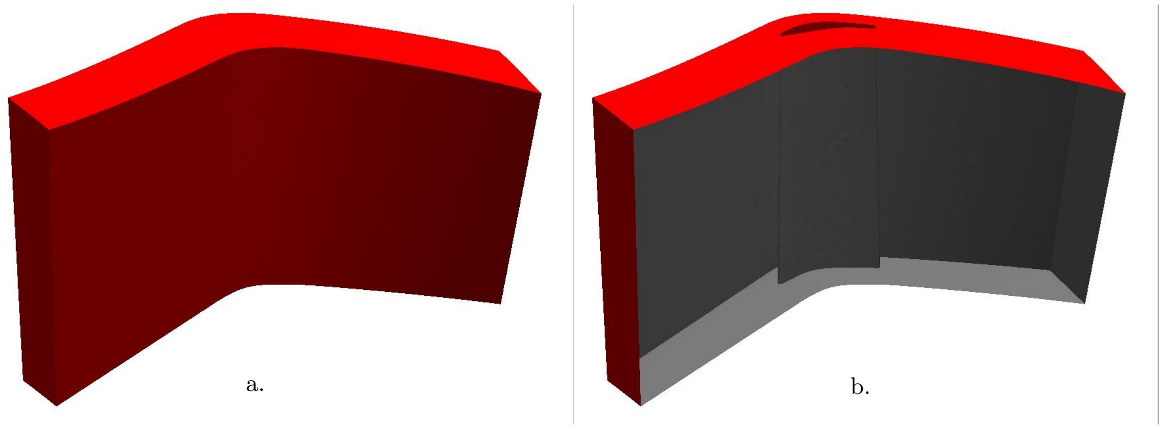


Figure 3: a) The fluid domain wedge, b) the fluid domain for CFD

design process).

There remain some issues that need attention before this view of design can be considered complete and realized. They are discussed below:

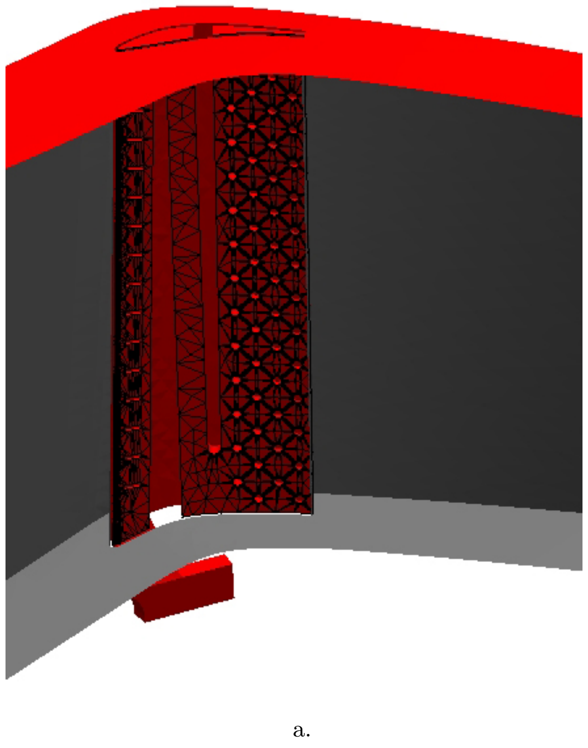
7.1 Regeneration and Consistency

The Master-Model method of generating parts has little ability to control the part's topological outcome. In fact, the resultant geometry may not be explicitly specified in any branch of the Feature Tree. This has the side effect that the topology may fundamentally change, even when two parts may be almost identical. Consider a wing/fuselage configuration where the fuselage is a cylindrical surface that due to the CAD modeler happens to be split along an axis generally aligned with the wing juncture. Assume the Master-Model has a parameter that controls the vertical mating position of the wing relative to the fuselage center-line, where zero aligns the wing with the seam of the cylinder. Consequently, the fuselage is maintained in CAD as 2 Faces each with half of the wing cutout. If the parameter is now set to a value greater than 1/2 of the wing thickness, the underlying topology is completely altered even though the parts differ only in a minor way. In this latter case, one of the two half cylinders is left untouched, while the other one ends up with a complete cutout (hole) where the wing fuses with the cylinder.

Topological inconsistencies are not a problem for most types of analysis, but can introduce difficulties in some cases. When using Adjoint methods (or other gradient approaches), one needs to compute geometric sensitivities for parameter changes. In order to perform this task, one determines the sensitivities either analytically or by computation. Because it is not possible to differentiate through the CAD system, the easiest way to get the sensitivities is to difference two instances. But, how does one track point movement from one part to a position in another instance when they differ at the topological level? Stated another way: how does one smoothly map from a start position to another position when components of the mapping can appear and disappear?

Assuming that there was some consistency at the topological level, tracking points is still problematic. The methods that provide a discrete view (i.e. triangulation) have no regard for history. All schemes place points to best satisfy some geometric criteria. Two parts that are almost identical will most likely display different tessellations, described by different densities of triangles. There has been some success with a technique that scribes a consistent quadrilateral patch topology over the CAD geometry [13].

It is possible to get sensitivities analytically if the construction is performed in a geometry kernel and control is exerted over the type of surfaces used. The result could be incorporated as an initial component of the Feature Tree. This has been done for turbomachinery aerodynamic design using Parasolid for construction[14]. UniGraphics could import the result of this work so that the fundamental blade shape could be defined outside of CAD as the rest of the model could be built upon the initial aerodynamic



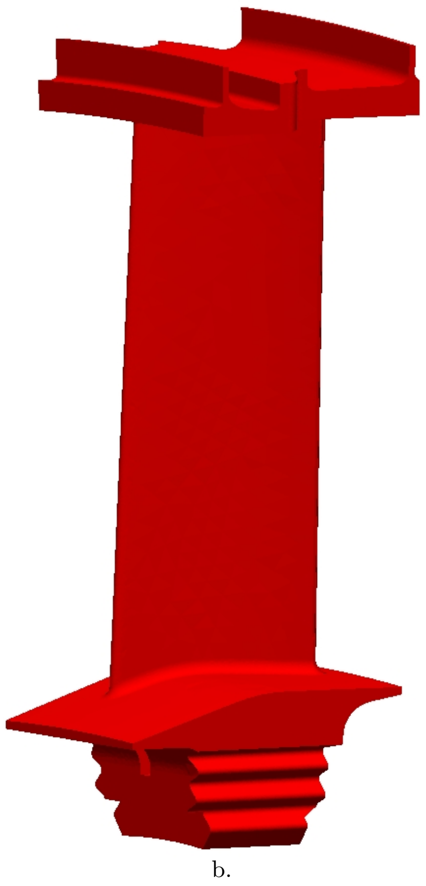


Figure 4: a) Complete flow regime, b) 75 blades in the row instead of 150.

blade shape. This would obvious make a more complex (and less consistent) system and detracts from the overall utility of the approach by limiting generality.

7.2 Tagging

In order to completely setup analysis codes, boundary conditions must be set. Some of the values are linked to the geometry (i.e. points of load application, surface(s) representing inflow, outflow or solid walls, etc.). Because the Feature Tree does not directly represent geometry, but creates the part, it is not clear from the tree perspective how to tag or map the resultant geometry.

This could be accomplished at the Face level by querying the CAD system for the branch of the Feature Tree that is responsible for generating the surface.

7.3 CAD Model Construction Issues

Top-down design is often referred to in CAD circles. All to often, proper techniques are not followed since proper practice requires slightly more effort up front in the design process. It is far too easy to create a bad one-off model that is virtually useless and must be thrown away if later changes are required. Engineers must be educated in the proper methodology for building robust CAD models that are extensible and can hold together through repeated design alterations and regeneration cycles.

7.4 Organizational changes

By far the most difficult challenge in having this approach adopted is not a technical one. Before the benefits articulated in this paper can be realized the process of design as it currently stands (in organizations that manufacture) needs to change. This is a difficult task in that the organization currently has a process that *works* no matter the efficiency and time-to-market for new designs.

ACKNOWLEDGEMENTS

The authors would like to thank the following CAPRI "usual suspects" for their prodding, assistance and/or support in this effort: Michael Aftosmis (NASA Ames Research Center), Juan Alonso (Stanford University), John Dannenhoffer III (Syracuse University), Peter Gage (NASA Ames Research Center), William T. Jones (NASA Langley Research Center), Al Magnuson (ICEM-CFD/ANSYS) and James Reuther (NASA Ames Research Center).

References

[1] R. Sampath, M. Kolonay and C.M. Kuhne. 2D/3D CFD Design Optimization Using the Federated Intelligent Product Environment FIPER Technology. AIAA Paper 2002-5479, September 2002.

[2] G. Follen, R. Claus, R. Blech, K. Meinert, D. Vandrei, A. Apel, V. Fields, R. Hare, N. Crawford, R. Ashleman, M. Davis, J. Reed. Numerical Propulsion System Simulation Architecture Definition. NASA TM 107343, November 1996.

[3] R. Braun. NASA's Intelligent Synthesis Environment Program: Revolutionizing The Agency's Engineering and Science Practice. Integrated Enterprise 2:1, Winter 2001.

[4] http://www.phoenix-int.com

[5] http://www.engineous.com

[6] http://www.cfdrc.com/products/ace

[7] http://www.omg.org/homepages/mfg/mfgcadv1-1rtf.htm

[8] http://www.transcendata.com/support/cadscript

[9] T.J. Tautges. Common Geometry Module: A Generic Extensible Geometry Interface. Proceedings of the 9^{th} International Meshing Roundtable. Sandia National Laboratories, October 2000.

[10] R. Haimes, and G. Follen. **C**omputational **A**nalysis **PR**ogramming **I**nterface. Proceedings of the 6^{th} International Conference on Numerical Grid Generation in Computational Field Simulations. University of Greenwich, June 1998.

[11] M. Aftosmis, M. Delanaye and R. Haimes. Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry. AIAA Paper 99-0776, January 1999.

[12] R. Haimes, and M. Aftosmis. On Generating High Quality "Water-tight" Triangulations Directly from CAD. Proceedings of the 8^{th} International Conference on Numerical Grid Generation in Computational Field Simulations. Honolulu HI, June 2002.

[13] J. Alonso, J. Martins, J. Reuther, R. Haimes and C. Crawford. High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model. Proceedings of the 16^{th} AIAA Computational Fluid Dynamics Conference, AIAA 2003-3429. Orlando FL, June 2003.

[14] A. Merchant and R. Haimes. A CAD-Based Blade Geometry Model for Turbomachinery Aero Design Systems. Proceedings of the 2003 ASME Turbo Expo, GT 2003-38305. Atlanta GA, June 2003.