CAD-Based Parametric Cross-Section Designer for Gas Turbine Engine MDO Applications

Christopher Dye[1], Joseph B. Staubach[2], Diane Emmerson[3] and C. Greg Jensen[4]

[1]UTC-Pratt & Whitney, christopher.dye@pw.utc.com
[2]UTC-Pratt & Whitney, joseph.staubach@pw.utc.com
[3]UTC-Pratt & Whitney, diane.emmerson@pw.utc.com
[4]Brigham Young University, cjensen@byu.edu

ABSTRACT

Because gas turbine engines are among the most complicated mechanical assemblies produced to date, there is an increasing need for computer-aided modeling programs that facilitate the incorporation of Multi-Disciplinary Optimization at the conceptual design stage. Here we report the development of Cross-Section Designer, a software tool for manipulating gas turbine engine geometry using the UGS NX CAD system [1]. Cross-Section Designer provides the user with an editable version of the engine gaspath based on output from a cycle specification program in the form of geometric and thermodynamic parameters. The system was designed to accommodate multiple parameterizations, and allows users to manipulate the geometry according to various different design schemes that best fit the design intent. The multiple-parameterization modeling methods described in this paper can be extended to the design of any system, providing real cost savings via design time reduction.

Keywords: CAD, Computer-Aided Design, Multiple Parameterizations, Optimization

1. INTRODUCTION

Gas turbine engine design is a complex activity, requiring multi-disciplinary modeling of the various component systems. This often includes the coordinated efforts of hundreds of designers whose engine design subspecialties include mechanical, aerodynamic, thermodynamic, durability, manufacturing, and maintenance disciplines. Design methodology of each discipline is applied in the creation of component systems called ‘modules’ within the engine.

Modular component design is managed at the system level by controlling the architectural and thermodynamic interfaces between modules, and at the module level by designing components to meet their interface requirements. In addition, customer requirements, manufacturing capability, and the ability to execute in a timely manner dictate the design direction and solution. This complex process is greatly facilitated by the use of computer-aided modeling programs. A good program should be both specific enough to provide meaningful models of specific processes or components, and flexible enough to be used in various application settings.

When modeling any system, it is essential to incorporate physics and geometry-based parameterizations that are representative of the underlying ‘real-world’ physics of the system’s aerodynamics, thermodynamics, and mechanical behavior. Peter Bartholomew [2] has outlined the need to adopt physics-based modeling techniques when modeling the overall mission performance of an aircraft in a multidisciplinary system. Physics-based parameterizations in the gas turbine design realm arise from a need to model flow through turbomachinery components. Codes are written to predict flow through given geometries, or to provide geometries to produce a given flow. The codes must be integrated with one another into a cohesive system in order to realize a Multi-Disciplinary Optimization (MDO) design environment, as shown in Townsend and Salas [3], as part of the development of an MDO design system for high-speed civil transport vehicles. When used in MDO, CAD and integrated codes running physics-based parameterizations can drive the design to a new, feasible solution based on customer requirements [4].

Ideally, methods for calculating performance based on design parameters must be integrated in a way that allows the designer to manipulate the design according to discipline-specific inputs. The methods described in this paper can be used in a general manner to operate using any parameterization and will aid in the creation of parameterization-based modeling methods applied to any product.

1.1 Low-Fidelity Modeling

It is important to distinguish between of low and high-fidelity component modeling. Low-fidelity models are used to investigate conceptual engine designs, while high-fidelity models define production-level engine part geometry. Low-fidelity models are more applicable than high-fidelity models for accomplishing the task of down-selection between architectures competing for the same requirements. They provide faster initial-approximations, which is preferable in design situations where little initial knowledge exists. In White et al. [5], a 1D modeling approach is preferred over higher-fidelity analyses of compressor flow for the aforementioned reasons. Low fidelity modeling must by necessity be rapid, allowing for changes in architectural topology according to the physical requirements of the problem and must conform to the time considerations of creating new concept models. Modeling at the low-fidelity level makes it possible to include new non-standard geometries with less modeling effort than creating full-detail, high fidelity models. Limitations include the inability to capture key detailed features that may prevent the design from ever materializing. This paper presents a low-fidelity-based method for improving the efficiency of aerodynamic modeling decisions.

Providing a more rapid method of design parameter specification can reduce design execution time. Often the inputs to automated modeling codes are so numerous that a user spends more time making perturbations to input files than the code takes to execute. For example, the cycle specification code used for this project, the flow meanline calculation “Flowpath”, takes roughly 30 seconds to calculate a full engine meanline cycle model, while the average user spends on the order of 5 to 60 minutes making simple changes to the input parameters. A program which is capable of manipulating the input geometry in a rapid, robust, physics-based manner would solve this problem and make it possible to iterate on geometric perturbations with the same accuracy and improved specificity.

1.2 Engine Cycle Modeling System

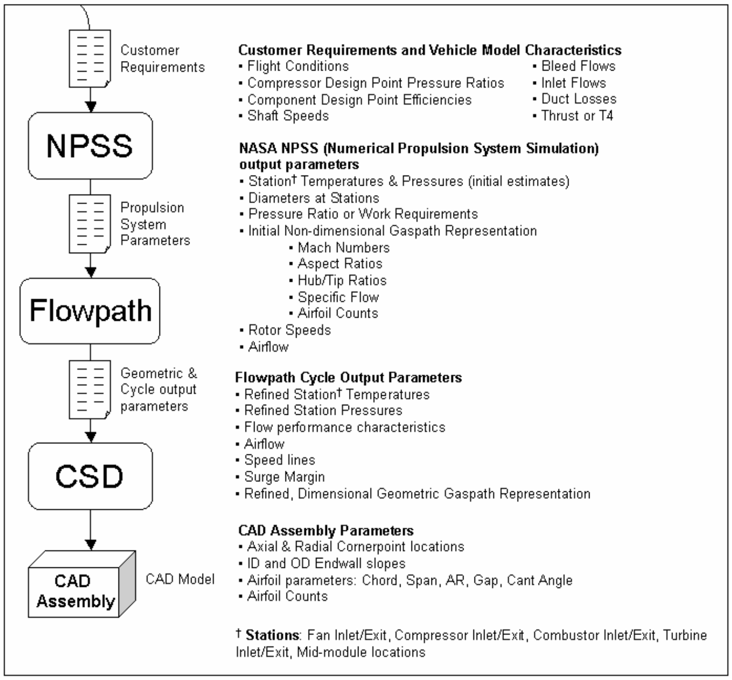


Fig. 1: Engine Cycle Modeling System Components and Inputs/Outputs.

The conceptual engine cycle modeling system of choice for Pratt & Whitney is a combination of computer codes, which specify a gas turbine engine cycle using thermodynamic and geometric parameters. As shown in Figure 1, inputs to the system are customer requirements in the form of vehicle performance characteristics.

Engine target performance characteristics include vehicle size, weight, operational conditions such as altitudes, distances, thrust requirements at various operating points, and target efficiency metrics such as fuel consumption rates. They are extracted from the customer inputs and used for initial propulsion system definition by NASA's NPSS program. NPSS calculates a best-fit engine type, specifying the cycle (high/low bypass ratio, number of turbines & compressors, number of airfoil stages in each turbine/compressor) based on an input of engine style (turboshaft, turbofan, etc.) and performance targets. The cycle definition serves as input to Flowpath, a one-dimensional meanline analysis, which calculates engine performance characteristics for the inner (core) flow stream and outer (bypass) flow stream of the engine. The meanline analysis takes as inputs the propulsion system parameters in the form of temperatures, pressures, mass flow, rotating component speeds, and basic geometric inputs such as diametral and axial (lengthwise) corner point positions, airfoil aspect ratios. The meanline analysis outputs refined values of the above inputs, including thermodynamic data, work output, dimensional and non-dimensional performance information, corner point positions, connecting duct lengths, profiles, etc. It is the geometric representation that will be discussed in this article. The geometric parameters are output in the form of airfoil geometry, duct shapes, and diameters/axial lengths at various stations along the engine axis.

2. MULTIPLE-PARAMETERIZATION MODELING METHODS

2.1 Cross-Section Designer

Cross-Section Designer (CSD) was created to address the geometry manipulation needs of an aerodynamic designer with an engine cycle definition code in the design loop. CSD is a CAD-native user application that allows the user to manipulate the basic engine geometry according to various parameterizations. CSD was written in ANSI C++ using standard libraries with CAD-modularity in mind. The program was written in two parts: a CAD-independent part consisting of C++ classes defining input/output routines, object-oriented (OO) data manipulation, and calculation algorithms; the second part was written using C and C++ languages and the UGS NX Open API (Application Programming Interface) to create GUIs (Graphical User Interfaces) and CAD geometry manipulation functionality. The GUIs and their operation will be discussed in this article, along with the parameterization algorithms, that reduce design cycle time significantly. These benefits arise because of the speed of the programmatic-parametric approach taken with this method, and other similar methods as in Rohm III and Tucker [8]. The parameterizations themselves are used widely throughout the jet engine industry to define geometry and thus the calculation logic in CSD is based on current industry practice. These standard parameterizations have been published in a field of scholarly articles, by White et al. [5], Beknev et al [6], Casey [7], to name a few. The parameterizations include the manipulation of engine gaspath geometry by changing airfoil sizes, aspect ratios (ratio of airfoil height, or span, to width, or chord), gaps in between airfoils, and areas at the inlet or exit of the engine section. The way that CSD combines these standard parameterizations into a central geometry manipulation tool is a novel approach. The incorporation allows a designer to change the engine geometry according to the desired method(s) in less time than applying the method to the airfoils or gaspath geometry manually.

2.2 CAD Representation of an Engine Gaspath

The working geometry for the CSD program is a CAD representation of a full-engine assembly. The CAD assembly contains the geometric information to specify the gaspath at a low-fidelity level. At this level of design, the pertinent information comes from the Flowpath meanline analysis and can be specified as a profile projection onto a 2D cross-section plane (See Figure 2). (The projection can be thought of as a “cut” through the engine, collinear with the engine's axis of rotation.) Airfoil corner points define the profile of blades and vanes, and splines define the shape of connecting ducts between modules. Airfoil corner points delineate the leading and trailing edge points at the root (hub) and tip of the airfoil. They are commonly referred to as ID (inner diameter) and OD (outer diameter) LE (leading edge) and TE (trailing edge) points. Corner point specification is the same for both static and rotating vanes, although the terms “root” and “tip” refer to OD and ID parts of a blade, and the ID and OD parts of a vane. Similarly, the fan bypass and engine core ducts are represented by inlet and exit points and slopes at the ID and OD. Splines of degree 3 are used to model the ducts, and airfoils are created using periodic NURBS.

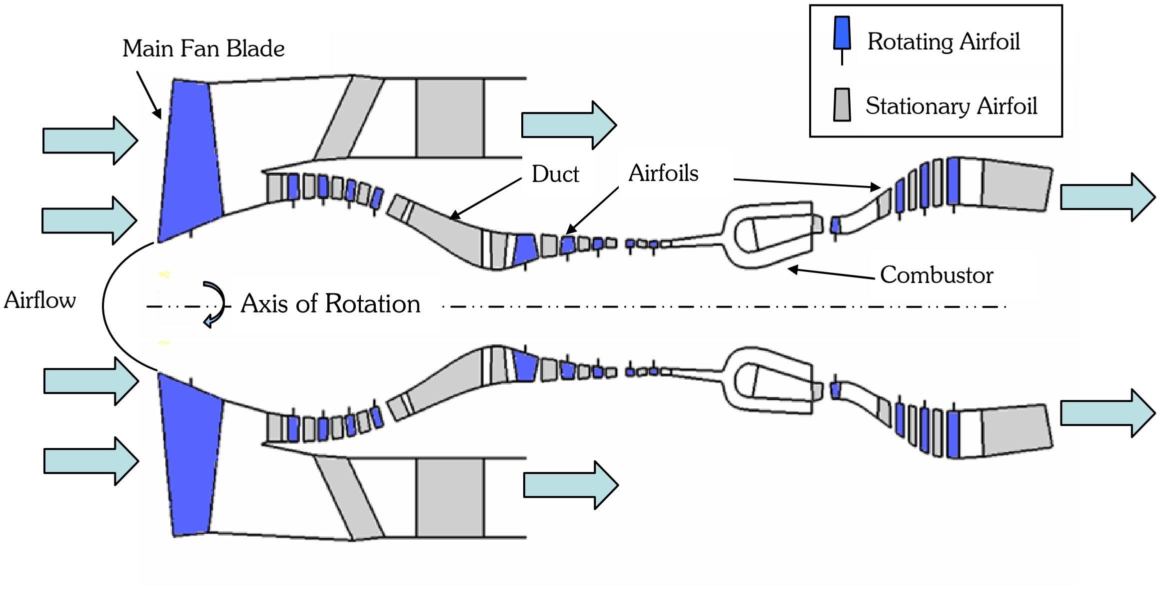


Fig. 2: 2D CAD representation of an engine gaspath: airfoils, ducts, and combustor. Gaspath profile shown from side; direction of airflow from left to right; airfoils represented as trapezoids.

A need arises for designers to propose and investigate adjustments to the airfoil and duct geometry in order to suggest design changes with discipline-specific intent. For example, consider a situation where a mechanical designer is selecting components to fit a high-speed engine. This engine gaspath may be very aggressive, resulting in high temperatures, overall pressure ratios (OPRs), and a flowpath that is small in ID radius. The designer may find that the flowpath is too small, in fact, to contain high-speed disks and a shaft that will accommodate high transfer torque. When the designer discovers this, he will need to adjust the engine flowpath by increasing the ID and OD radii while holding important key aerodynamic attributes like inlet and exit areas, airfoil aspect ratios, axial gaps, etc. After the flowpath has been adjusted to accommodate the mechanical components, the designer will need to verify that the engine design will meet performance requirements by re-executing the meanline analysis to substantiate the corner point locations.

The design problem that CSD solves is to manipulate a gas turbine flowpath according to discipline-specific inputs. When designers want to make modifications to a flowpath, they necessarily do so from a discipline-specific perspective, changing the geometry of the engine according to a parameterization that fits the intent. As in the above example, the designer chose to adjust the geometry directly to influence mechanical clearances, and then re-run the meanline analysis to verify the geometry integrity. The designer could just as easily have chosen to change inlet or exit area of the compressor, which would necessitate a change in rotational speed, and a recalculation of the meanline analysis. CSD incorporates a number of parameterizations that operate on the same underlying flowpath geometry. As discussed in later sections, it is capable of manipulating the gaspath and airfoils to change flow area, airfoil corner-point geometry, and airfoil gaps in a physics-based manner.

2.3 Module-based Parameterization

The Module-based (mod-based) parameterization manipulates the airfoil corner points in a way that is fitting to the design of axial compressors and turbines. The method is particularly well suited to compressors with high radius change from inlet to exit, requiring smooth continuity between adjacent airfoils. The mod-based parameterization creates splines representing the ID and OD flowpath, onto which the corner point locations are placed. See Figure 3. The airfoil corner point locations are placed by calculating the axial chord of each airfoil gap between each adjacent airfoil, from front to back within the module. The mod-based parameterization is capable of quickly establishing smooth continuity between adjacent airfoil corner points over a module of large axial span, making it most useful for low-speed, low-pressure compressors and turbines.

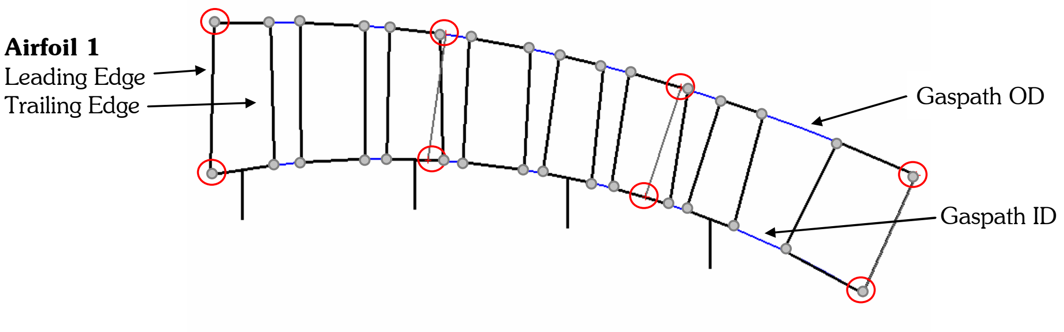


Fig. 3: Module-based Parameterization Applied to Low Pressure Compressor Airfoils Blue Curves: ID and OD Gaspath Splines; Red Circles: Spline Defining Points (4 ID, 4 OD); Gray Dots: Airfoil Corner Points (4 Per Airfoil).

The module-based algorithm performs the following steps:

1. Creates splines defining the ID and OD of the gaspath.
2. Creates lines of infinite length representing the first airfoil leading and trailing edge, based on axial chord and aspect ratio (first airfoil corner point is at axial zero position).
3. Intersects the lines to determine corner points at ID/OD LE/TE.
4. Calculates next airfoil leading edge position based on axial gap value to next downstream airfoil.
5. Repeats steps 2-4 for all airfoils in module.

2.3.1 Spline Mathematics

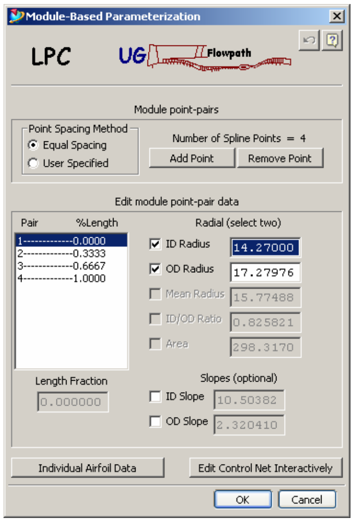
The module-based parameterization uses Bezier spline math to compute the profiles of the ID and OD gaspath. Splines are best-fit cubic using uniform knot-vectors, computed using the following equations for general 3rd order uniform Bezier curves [9]:

$$S_i(t) = \sum_{k=0}^{3} P_{i-3+k} b_{i-3+k,3}(t), \qquad t \in [0,1] \tag{2.0}$$

$$S_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} 1 & 3 & -3 & -1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix} \tag{2.1}$$

2.3.2 User Adjustments to Flowpath

The user may choose to adjust the entire module via the ID/OD splines or to adjust the individual airfoils. The module may be adjusted by changing the spline defining points' axial and radial locations, increasing the number of spline-defining points, modifying ID and OD inlet and exit parameters such as slope, radius, mean radius, radius ratio, or flow area. Choosing the individual airfoil editing option executes the row-based menu for access to the individual aerodynamic parameters such as gap, chord, aspect ratio, etc. and when the user has completed the adjustments the program returns to the Module-based menu. (Figure 4)



Module point-pairs

Point Spacing Method
● Equal Spacing
○ User Specified

Number of Spline Points = 4

Edit module point-pair data

Pair	%Length
1	0.0000
2	0.3333
3	0.6667
4	1.0000

Radial (select two)

☑ ID Radius 14.27000
☑ OD Radius 17.27976
☐ Mean Radius 15.77488
☐ ID/OD Ratio 0.825821
☐ Area 298.3170

Length Fraction 0.000000

Slopes (optional)

☐ ID Slope 10.50382
☐ OD Slope 2.320410

Fig. 4: Module-based Parameterization Menu.

2.3.3 Editing Spline Defining Points

Spline defining points may be edited by adding a new number of points and/or changing the distribution of the defining points within the module. Increasing the number of spline-defining points increases the order of the polynomial, allowing greater oscillation of the flowpath wall, and decreasing the number of points likewise reduces the order. When the splines are edited without changing degree, the curvature is re-computed using Bezier spline blending function, Eq. 2.0. If the spline degree is edited, a similar higher-degree Bezier blending function is used. Additional blending functions may be found in Bezier mathematics texts. As splines are edited, airfoil geometry is recalculated to fit the splines and the module is reconstructed to fit the Flowpath, as shown in the Figure 5.

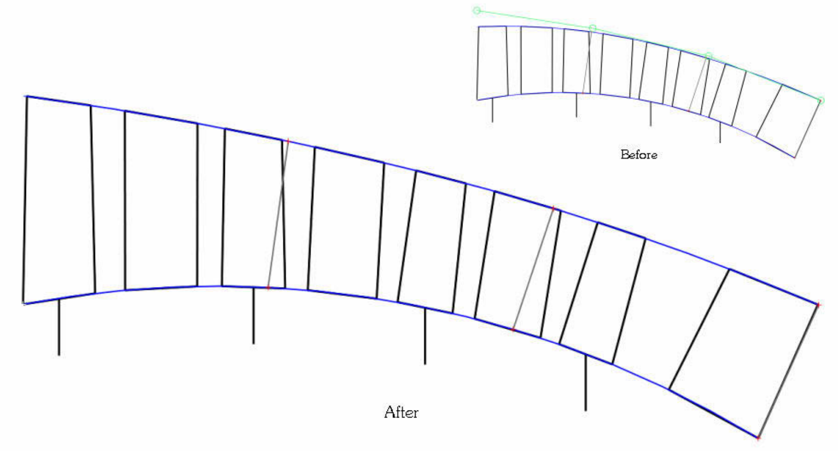
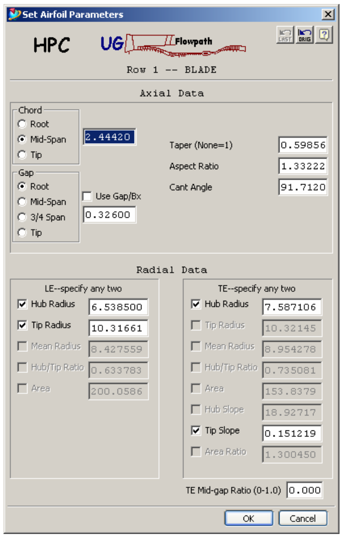


Fig. 5: Area change to module via radius increase of OD Spline-defining point #1.

2.4 Row-based Parameterization

The row-based parameterization allows the designer to alter airfoils individually or as pairs (stages). Blades and vanes are defined using traditional airfoil aerodynamic parameters: axial chord, span, aspect ratio, etc. These parameters are common to the gas turbine industry and provide a means of specifying a 2D representation of an airfoil cross-section. See Figure 6 below for the Row-based Parameterization GUI.



Row 1 -- BLADE

Axial Data

Chord
○ Root
● Mid-Span
○ Tip
2.44420

Gap
● Root
○ Mid-Span
○ 3/4 Span
○ Tip
☐ Use Gap/Bx
0.32600

Taper (None=1) 0.59856
Aspect Ratio 1.33222
Cant Angle 91.7120

Radial Data

LE--specify any two
☑ Hub Radius 6.538500
☑ Tip Radius 10.31661
☐ Mean Radius 8.427559
☐ Hub/Tip Ratio 0.633783
☐ Area 200.0586

TE--specify any two
☑ Hub Radius 7.587106
☐ Tip Radius 10.32145
☐ Mean Radius 8.954278
☐ Hub/Tip Ratio 0.735081
☐ Area 153.8379
☐ Hub Slope 18.92717
☑ Tip Slope 0.151219
☐ Area Ratio 1.300450

TE Mid-gap Ratio (0-1.0) 0.000

Fig. 6: Row-based Parameterization Menu.

2.4.1 User Adjustments to Flowpath

With a total of 25 possible airfoil parameters, the row-based parameterization employs a logical algorithm that allows the user to adjust the airfoil by inputting coordinating combinations of parameters. By specifying a limited number of essential parameters, the remaining unknown parameters may be calculated using the known parameters. This provides flexibility of design for manipulation of airfoils using multiple input parameterizations. See the following set of figures (Figures 7-8) for the complete list of parameters used in the Row-based parameterization:

2.4.1.1 Axial & General Parameters

a) Chord at Root
b) Chord at Mid-span
c) Chord at Tip
d) Gap at root
e) Gap at Mid-span
f) Gap at ¾ span
g) Gap at tip
h) Cant Angle
Taper (c/a)
Aspect Ratio [(Tip Radius-Hub Radius)/a]

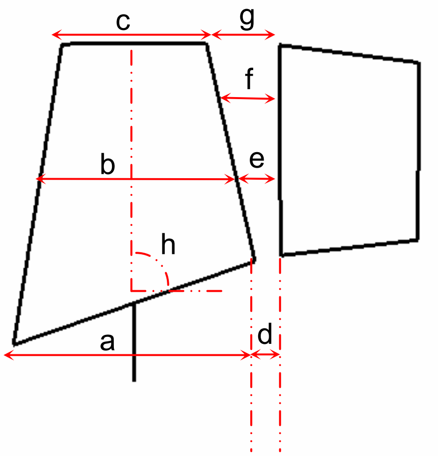


Fig. 7: Axial Row-Based Parameters.

2.4.1.2 Radial Parameters

Leading Edge (LE)

Any two of the following:

a) Hub radius
b) Tip radius
c) Mean radius
Hub/tip ratio (a/b)
Area

AND

Trailing Edge (TE)

Any two of the following:

d) Hub Radius
e) Tip Radius
f) Mean Radius
g) Hub Slope
h) Tip Slope
Area Ratio
Hub/Tip Ratio (a/b)
Area

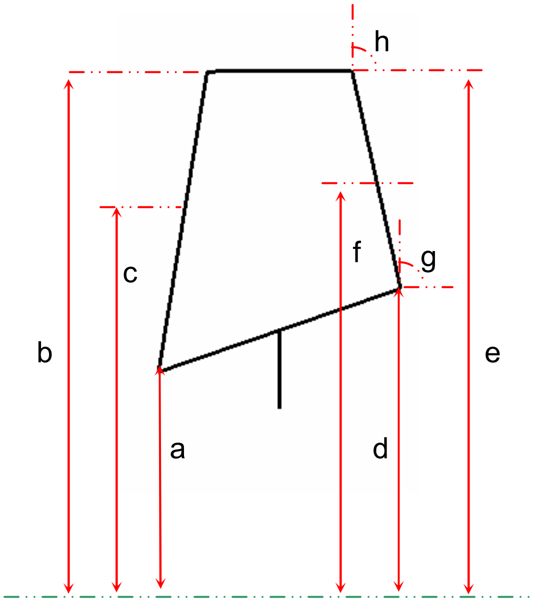


Fig. 8: Radial Row-Based Parameters.

2.5 Scaling Parameterizations

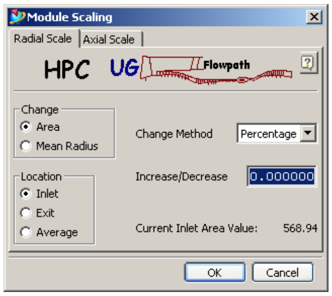
CSD employs two scaling parameterizations for manipulating flowpath geometry by variations on geometric, or photographic, scaling. These parameterizations provide the ability to make axial or radial changes to the airfoil corner points. Axial and radial scaling operations adjust the airfoil corner points based on modifying areas and mean radii, for the radial parameterization, and modifying gap and chord values for the axial parameterization.

2.5.1 Radial Scaling

Changes to the radial attributes of the gaspath section may be made via an increase or decrease to either the area or mean radius at an axial location within the module. Area or mean radius may be changed at the inlet, exit, or at the mean axial location of the flowpath. As seen in Figure 9, the parameterization allows the user to choose to distribute the area change as a percentage of current area.

2.5.2 Axial Scaling

The axial scaling parameterization provides the designer with the traditional method for manipulating airfoil axial values. The user may input a new axial length (or percentage of current length in the same manner as the radial parameterization) and choose to distribute the length delta across the gaps only, the airfoil chords only, or both airfoil gaps and chords. This parameterization is used most extensively in length-reduction activities. (Figure 10)



Change
● Area
○ Mean Radius

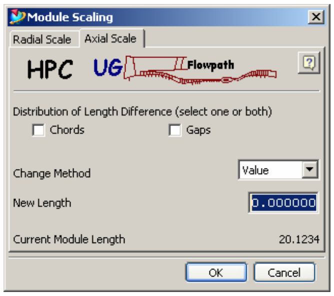
Location
● Inlet
○ Exit
○ Average

Change Method: Percentage

Increase/Decrease: 0.000000

Current Inlet Area Value: 568.94

Fig. 9: Radial Scaling Parameterization Menu.



Distribution of Length Difference (select one or both)
☐ Chords ☐ Gaps

Change Method: Value

New Length: 0.000000

Current Module Length: 20.1234

Fig. 10: Axial Scaling Parameterization Menu.

2.6 Inter-module Duct Parameterization

Connecting ducts are modeled within the program using Bezier curves of degree 3 (user-editable to increase degree and profile), maintaining connectivity and tangency (knot conditions G0 and G1) at endpoints connecting to adjacent modules. The Duct Parameterization allows the user to insert intermediate points at positions along the duct to alter the profile and design the expansion/contraction rate of the duct according to basic aero parameters including area and station along the flow wise direction. See Fig. 11 for an example of area increase using the duct parameterization.

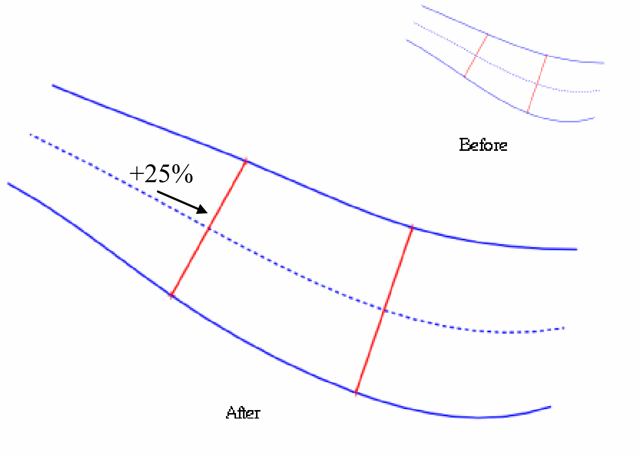


Fig. 11: Duct area increased at 25% streamwise location via control surface #1 (red line on left).

3. CONCLUSIONS

The Module-based parameterization, using degree-three Beziers provides a good representation of high-curvature modules and has been demonstrated to accurately represent modules from over 30 distinct engine gaspaths in the Pratt & Whitney modeling system. A user can make changes such as area ratio and curvature adjustment in just a few minutes, allowing for more rapid iterations on gaspath geometry than ever before.

The Row-based parameterization is used to make changes to individual airfoils for part sizing and clearance needs. Perturbations can be made in a few seconds to the airfoil geometry, resulting in a reduction of model adjustment time. Modules of airfoils can be adjusted according to proven acceptable parameterizations. Designers may choose to re-gap a module of airfoils at a time, in a few seconds. Increases in blade size can alter airfoil work split throughout the module by adjusting airfoil chords. This parameterization has been used most often in an effort to reduce module or engine length, reducing weight and increasing efficiency.

The inter-module duct parameterization benefits the designer by allowing contour changes within the module to be made while holding aerodynamic parameters constant, in an effort to make perturbational geometric improvements while maintaining low duct losses.

In summary, these parameterizations provide a useful tool for making adjustments to fundamental engine gaspath geometry at a conceptual, low-fidelity level. Coupling an improved modeling iteration scheme with the ability to associate these geometric perturbations with parametric perturbations of higher-fidelity parts creates a system that rapidly responds to changes and has had far-reaching results at Pratt & Whitney. Further work is ongoing to associate 3D part models to these 2D representations in order to make quick, associative, parametric changes to full-engine models based on discipline-specific inputs. Models can be created associatively by passing parameters from the CSD parameterization tool to models built with coordinating parameterizations, updating higher-fidelity CAD models. This pushes geometry associativity towards the goal of Blair and Reich [10] in a "fully-associative aerospace design environment". See Fig. 12 for a view of a 2D-3D associative gas turbine engine CAD model.

These methods may be applied to the design of any mechanical system in any industry. Parameterizations specific to an industry may be incorporated into a CAD-based interactive design system allowing the user to manipulate the design geometry according to discipline-specific inputs. One can anticipate similar benefits, if incorporated correctly, in the form of significant time reductions and faster convergence to a feasible design solution.

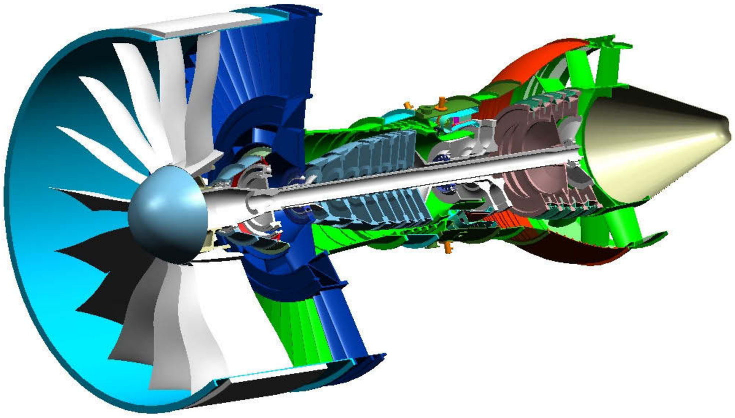


Fig. 12: CAD Model created with associative CSD multiple parameterization methods.

4. ACKNOWLEDGEMENTS

The authors would like to acknowledge the additional core developers of the CSD tool. BYU Mechanical Engineering students Hans Soderquist, Damon Delap, Matt Wilson, Benjamin Dunn, Matt Fisher, and Pratt & Whitney Designer Robert Kline are responsible for the robust functionality and flexibility of the tool. Hans Soderquist provided code algorithms for the inter-module duct parameterization as well as the overall CAD assembly of the modules. Damon Delap created the first versions of the Menu GUIs and with Matt Wilson wrote the logic foundation for the Row-based and Module-based parameterizations. Matt Wilson and Rob Kline are credited for their architectural vision of the overall object-oriented code structure. Ben Dunn provided object-oriented program development and flexible user-interactive selection and airfoil data storage. Matt Fisher made additions to the Global and Axial Scaling routines.

5. REFERENCES

[1] UGS NX is a licensed product of UGS Corporation, a division of EDS (http://www.eds.com).

[2] Bartholomew, P.: Defence Evaluation and Research Agency, Farnborough, Hampshire, UK: The Role of MDO Within Aerospace Design and Progress Towards an MDO Capability, AIAA-98-4705.

[3] Townsend, J. C.; Salas, A. O.: Managing MDO Software Development Projects, American Institute of Aeronautics and Astronautics, AIAA-2002-5442.

[4] Samareh, Jamshid A.: Computer Sciences Corporation, Geometry Laboratory (GEOLAB): Use of CAD Geometry in MDO, 6th AIAA/USAF/NASA/ISSMO Symposium on Multi-Disciplinary Analysis and Optimization", AIAA-96-3991.

[5] White, N. M.; Touridakis, A.; Elder, R. L.: Axial compressor performance modeling with a quasi-one-dimensional approach, Proc Institution of Mech Engineers, Vol 216, Part A: Power & Efficiency, 181-193.

[6] Beknev, V. S.; Egorov, I. N.; Talyzina, V. S.: Multicriterion Design of the Optimization of the Multistage Axial Flow Compressor, ASME Cogen-Turbo IGTI-Vol. 6, 453-460.

[7] Casey, M. V.: A mean line prediction method for estimating the performance characteristic of an axial compressor stage, Sulzer Escher Wyss Limited, Zurich, Switzerland, C264/87.

[8] Rohm III, T.; Tucker, Scott S.; Jones, Chris L.; Jensen, C. Greg: Parametric Engineering Design Tools and Applications, Proceedings of ASME Design Automation Conf. Sep 10-13, 2000, Baltimore, MD DETC2000/DAC-14275, p657-664

[9] Sederberg, Thomas W.: Computer-Aided Geometric Design, Ch.1, 1997. http://students.cs.byu.edu/~tom/557/text/ch1.pdf

[10] Blair, M.; Reich, G.: A Demonstration of CAD/CAM/CAE in a Fully Associative Aerospace Design Environment, AIAA Paper 96-1630, Apr. 1996.