

A Spline-Based Modeling Algorithm for Application to Aerodynamic Shape Optimization Based on CFD Analysis

Kisun Song, Kyung Hak Choo, and Dimitri Mavris
 Georgia Institute of Technology

ABSTRACT

In early phases of conceptual design stages for developing a new car in the modern automobile industry, the lack of systematic methodology to efficiently converge to an agreement between the aesthetics and aerodynamic performance tremendously increases budget and time. During these procedures, one of the most important tasks is to create geometric information which is versatilely morphable upon the demands of both of stylists and engineers. In this perspective, this paper proposes a Spline-based Modeling Algorithm (SMA) to implement into performing aerodynamic design optimization research based on CFD analysis. Once a 3-perspective schematic of a car is given, SMA regresses the backbone boundary lines by using optimum polynomial interpolation methods with the best goodness of fit, eventually reconstructing the 3D shape by linearly interpolating from the extracted boundaries minimizing loss of important geometric features. As a preliminary study, an aerodynamic shape optimization based on CFD analysis is conducted using ANSYS fluid dynamics solutions. To maximize the performance, we employed state-of-art design methodologies such as design-of-experiments (DOE) and surrogate modeling techniques that incorporate a series of procedures from creating geometries to conducting CFD simulations. The SMA is programmatically integrated with the CFD procedures to automatically morph and create corresponding automobile external shapes. For validation and verification, three notional vehicles are explored by investigating aerodynamic performance. Not only does the baseline geometries obtained from SMA show well-matched drag coefficients (C_D) but also the optimization study results in a significant amount of C_D reduction.

CITATION: Song, K., Choo, K., and Mavris, D., "A Spline-Based Modeling Algorithm for Application to Aerodynamic Shape Optimization Based on CFD Analysis," *SAE Int. J. Passeng. Cars - Mech. Syst.* 10(2):2017, doi:10.4271/2017-01-1510.

INTRODUCTION

The advent of computer-aided-design (CAD) and computer-aided-engineering (CAE) techniques in recent decades has brought about a paradigm shift in designing a new car in the modern automotive industry. Numerous innovative approaches have enabled designers and engineers to make and change their decisions much more efficiently than before, and have significantly reduced time and cost of the design process. Particularly in the conceptual design stage, tools such as computational fluid dynamics (CFD) and advanced CAD have accelerated these research processes tremendously. These tools have enabled stylists and aerodynamic engineers to exploit fully the synergism of collaborative interactions with each other to obtain optimized (qualitatively as well as quantitatively) external shapes that satisfy both aesthetics and aerodynamic performance simultaneously. Via these endeavors, the primary focus in designing a car has shifted from just "designing" or "creating" to "optimizing."

Therefore, securing the reproducibility of geometric information and how to efficiently iterate design processes have become important. In this perspective, some adopted commercial engineering automation tools. Thanks to these, aerodynamic engineers could manage other tasks during the jobs are being automatically taken care of. Takiguchi et al.[1] developed a tool that forms exterior shapes from vehicle specifications, generates mesh, conducts CFD calculations, and

visualizes results in an automated manner. Ando et al.[2] created an environment that successively performs morphing, meshing, CFD calculations, and response surface generation for aerodynamic evaluation at the conceptual stage of automotive styling.

Others developed and applied numerous advanced design space exploration methodologies have been implemented to retrieve optimal results with minimal investments. A minimized number of trials are created by design-of-experiments (DOE), and the corresponding results are approximated by surrogate models. Then, additional CFD analyses can be avoided and replaced by investigating the model's behavior. Song et al.[3] developed an aerodynamic design optimization methodology to minimize drag based on CFD analysis using DOE and artificial neural networks (ANN). The authors also performed various parametric studies to demarcate important design variables and note negligible ones associated with an external shape of a passenger sedan.[4][5][6] Kang et al.[7][8] and Cho et al.[9] conducted a CFD-based parametric study to investigate the effects of various actively translating rear diffuser devices on the aerodynamic performance of a fast-cruising passenger car. Kim et al.[10] performed optimization studies of internal engine room flow using aero-thermal CFD analysis. In fact, DOE and surrogate modeling techniques are widely accepted in various engineering conceptual design fields. Jeong et al.[11]

proposed an aerodynamic design optimization of wind turbine blades under turbulent winds. Choo et al.[12] generated response surfaces from DOE results to facilitate the development process of the prediction model of particulate matter in the conceptual design of aircraft engines.

The aforementioned accomplishments, in common, have needed a design methodology to create geometry with quality and efficiency to integrate with the aerodynamic design processes. Naturally, various external shape design methodologies have been developed and proposed. Rho et al.[13] developed a vehicle modeling function (VMF) for three-dimensional shape optimization. The author decomposed a generic passenger vehicle shape and modeled the outer boundary lines by using a series of section functions based on Bernstein and Bezier curves. Calkins et al.[14] dealt with a parameterized shape design method known as knowledge-based design or rule-based design to provide a genuine prototype of a product ahead of committing to manufacturing. Even recently, 3D car scanning technology[15] is widely used in the automotive industry.

As a matter of fact, however, creating a geometry is a trade-off decision making problem between quality and cost. If categorized, the former[13][14] studies are pursuing the efficiency with minimum efforts and costs whereas the latter[15] is invest in time and cost to obtain very detail car geometric information. Therefore, the relevant research question becomes as follows. Can there be a compromised solution to create vehicle geometry between geometric quality and cost? To what extent of the quality should be captured in pursuing modeling efficiency? Based on what approach should the method be expected to perform its task? In response to these questions, this paper introduces a Spline-based Modeling Algorithm dubbed SMA, an algorithm that reconstructs a 3D geometry from three-perspective schematic geometry source data. Once a three-perspective schematic of a car is given which consists of views of top, side, front, and rear, SMA regresses the backbone boundary lines by using the optimum polynomial interpolation method with the best goodness of fit based on parameterizing the width, length, and height in Cartesian coordinates system. From each schematic, two-dimensional geometric information can be obtained and one of the three axes is set to be the independent variable, having the rest two axes become the dependent of the function of it. Depending on the research's choice, the correlation among the three axes can be freely determined. Inspired by the discovery that the shape of passenger vehicles is smooth and continuous within the surfaces defined by corresponding boundary lines, SMA provides three different spline options SMA: linear, cubic spline, and Piecewise Cubic Hermite InterPolation (PCHIP). Once the backbone boundary lines are captured, the 3-D surfaces are eventually reconstructed by using linearly interpolating from the boundaries with minimum loss of the detail geometric features and maximum efficiency. In overall SMA not only provides detailed external vehicle shape but also parameterized information of every geometric features which will allow the researchers to morph the shapes and potentially be harmonious to be implemented into performing numerous aerodynamic design optimization studies.

The rest of this paper includes the following topics. In the methodology section, detailed descriptions of SMA are provided. Next, for validation and verification of the performance of SMA, the resultant baseline geometries of several cars undergo CFD analysis, and the corresponding drag coefficients (C_D) are investigated with reference data. The three explored notional car shapes are like the Tesla Model S, the Infiniti G37 coupe, and the Audi A6, respectively. Finally, an aerodynamic design optimization to minimize C_D with four geometric variables of the modeled G37 coupe is conducted as a preliminary integrated design optimization workflow to fully harmonize SMA with ANSYS Workbench in the future research.

METHODOLOGY

Mathematical Background

A Short Review on Polynomial Spline Interpolations

The first procedure is to extract the backbone boundary lines, which will be exoskeletal frames of the modeled car. Once a designer determines the boundary lines from the input schematics, SMA will choose the most appropriate spline function form out of three spline options: cubic, piecewise cubic Hermite, and linear.

A cubic spline consists of third-order polynomials, each of which is constructed at every interval. Compared to the polynomial interpolation, spline interpolation can avoid the problem of Runge's phenomenon, where oscillation can occur between points when interpolating using high-degree polynomials. The basic form of a one-dimensional cubic spline interpolation between point i and $i+1$ is shown in [equation 1](#) ($i=0\sim n-1, t_i \leq t \leq t_{i+1}$).

$$f_i(t) = a_i + b_i(t-t_i) + c_i(t-t_i)^2 + d_i(t-t_i)^3 \quad (1)$$

For n data points, there are $n-1$ intervals and thus $4(n-1)$ unknowns to evaluate to solve all the spline function coefficients. For solving the coefficients, three boundary conditions are used; the function values, first-order derivatives, and second-order derivatives of two adjacent splines at point $i, f_i(t_{i+1})$ and $f_{i+1}(t_{i+1})$, must be continuous:

$$f_i(t_{i+1}) = f_{i+1}(t_{i+1}) \quad (2)$$

$$f'_i(t_{i+1}) = f'_{i+1}(t_{i+1}) \quad (3)$$

$$f''_i(t_{i+1}) = f''_{i+1}(t_{i+1}) \quad (4)$$

These conditions yield $4(n-1)-2$ total equations, and the remaining two equations are obtained by applying a boundary condition at points t_0 and t_{n-1} , one at each.

A piecewise cubic Hermite spline uses the same basis function form as in [equation 1](#). The difference is that it does not enforce the condition described by [equation 4](#); the second derivatives of neighboring spline functions must be continuous. Despite all good qualities, a cubic spline interpolation gives, depending on the behavior of points, the spline may not be able to provide the correct shape for an interpolant in an application. For this reason, a piecewise cubic Hermite spline is used; it allows more user control by relaxing the described conditions above. To specify the unique set of coefficients, Fritsch et al.[16] developed an approach, called monotone Hermitian interpolants, which implements the following requirements.

1. At each control point, “reasonable” derivatives are estimated.
2. The derivative is zero if the control point is a local maximum or minimum.

These two requirements guarantee that the interpolation will be locally monotone. More specifically, if the function values at two different control points are the same, then the interpolation for the interval is monotone. Last, a linear spline consists of linear functions established by [equation 5](#).

$$f_i(t) = f_i(t_i) + \frac{f_i(t_{i+1}) - f_i(t_i)}{t_{i+1} - t_i} (t - t_i) \quad (5)$$

Therefore, as function values at all control points are known, no further calculations are needed. To illustrate the general shapes of the aforementioned spline interpolations, sample x and y sequences are created: $x = [0 \ 1 \ 2 \ 3 \ 4 \ 5]^T$, and $y = [0 \ 0 \ 3 \ 4 \ 5 \ 5]^T$. Corresponding graphs are illustrated in [Figure 1](#).

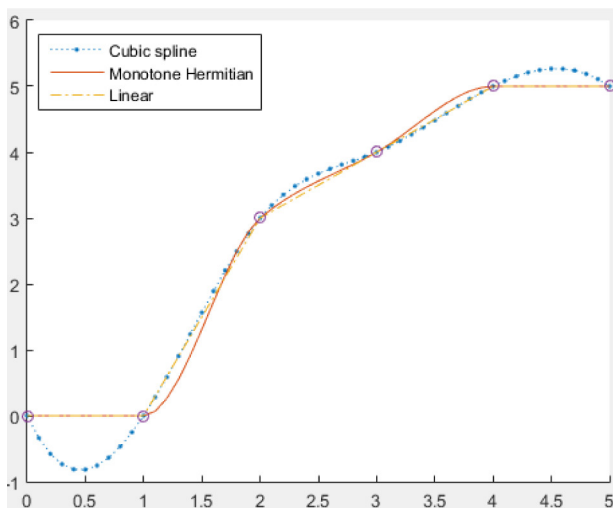


Figure 1. Comparison of several spline interpolation functions

Regression of Backbone Boundary Lines

Let there be a chosen backbone boundary line from the 3D schematic geometry, as highlighted in [Figure 2](#), whose shown illustration is a part of the top and side schematics of the Audi A6[17].

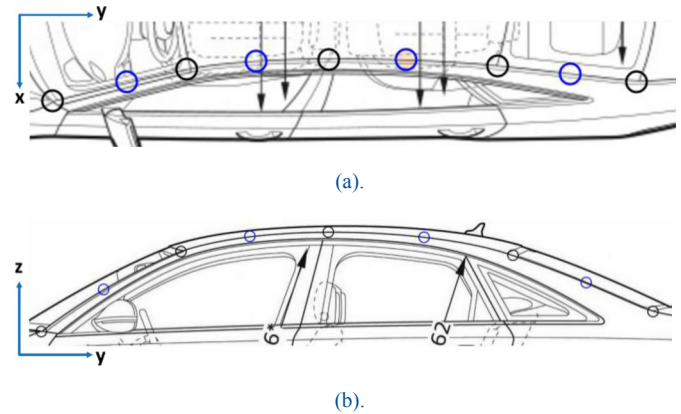


Figure 2. Example showing backbone boundary points (a) top and (b) side

Let black hollow circles be control points and blue hollow circles be points to be used to perform a model representation error (MRE), which is a method of evaluating the goodness of fit of the spline by evaluating the error between the interpolation value and the true value at the points that are not used to conduct regression. The directions of x , y , and z axes are also depicted: x originates from the center of the car to the side, and y originates from the foremost point toward the rearmost point of the car. Finally, z is from the ground and deployed upward.

From [Figure 2\(a\)](#), x and y coordinates corresponding to the shoulder line that passes through the A, B, and C pillars are regressed. Additionally, from [Figure 2\(b\)](#), y and z coordinates are obtained. Intuitively, y is the parameterized axis because it continues to increase monotonically in both cases (from the front to the rear). Therefore, by using a Piecewise Cubic Hermite Interpolation (PCHIP), x , y , and z are regressed for the black hollow circles. Because y is the parameterized axis, $y(t) = t$. Therefore, by feeding this into the regression formulas, f_1 and f_2 , we can eventually obtain the parameterized equations of x , y , and z describing the given backbone lines in the 3D space, as summarized in [equation 6](#).

$$\begin{aligned} y(t) &= t \\ x(t) &= f_1(y(t)) = f_1(t) \\ z(t) &= f_2(y(t)) = f_2(t) \end{aligned} \quad (6)$$

Finally, three sets of x , y , and z are created by three spline options. Subsequently, SMA determines the best-matching spline options by investigating corresponding MREs at the blue-colored circles in [Figure 2](#). The process shown in [Figure 2](#) is only for one line. Therefore, SMA will be used to repeat these tasks for all identified backbone boundary lines. These tasks are represented in the following pseudo-code.

```

Import backbone boundary point datum ( $l_1, l_2, \dots, l_n$ )
REG=[]
for  $i=1$  to  $n$ 
    R=[]; MRE=[]
    for  $j=\text{each of [cubic spline, PCHIP spline, linear spline]}$ 
         $R_j \leftarrow \text{Regress } l_i(x(t), y(t), z(t)) \text{ using interpolation method } j$ 
         $MRE_j \leftarrow \sum \left( (R_{jx} - R_{ix})^2 + (R_{jy} - R_{iy})^2 + (R_{jz} - R_{iz})^2 \right)$ 
    end
     $REG_i \leftarrow R(x(t), y(t), z(t)) \text{ having the minimum MRE}$ 
end

```

Algorithm 1. Backbone Boundary Lines Regression Algorithm

Where R_{jx} means the true x value of the blue circle in Figure 2, whereas R_{ix} represents the regressed x value obtained by using interpolation method j , and vice versa for y and z . Figure 3 is an illustration of the regression results of the Audi A6. Line 1 in Figure 3 is the line considered in Figure 2. SMA chose the PCHIP method in regressing this line.

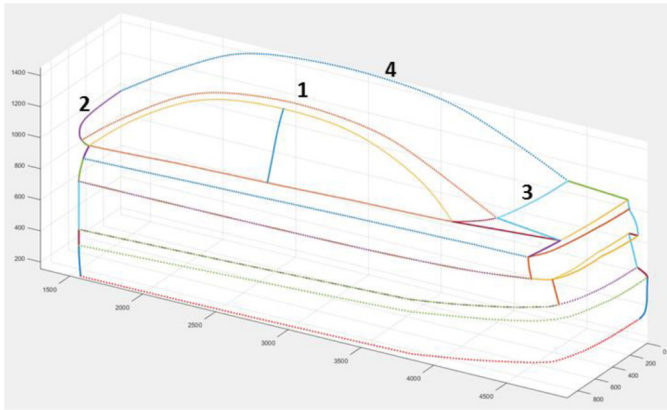


Figure 3. Example of regressed backbone boundary lines of the Audi A6

Constructing External Surfaces from Regressed Lines

Surface Construction by Linear Differential Interpolation

Now all the surfaces can be constructed out of the data retrieved from the algorithm 1. In a numerical calculation, qualitatively, the simplest function is a linear function. Moreover, the simplest arithmetic operations are addition and subtraction. Therefore, SMA is developed to reconstruct surfaces through a combination of additions and subtractions. For explanation, consider that we would construct a surface enclosed by the lines from 1, 2, 3, and 4 in Figure 3. Let us call this surface as S . Overall terminology conventions are represented in Figure 4, which is the isometric view of the surface S .

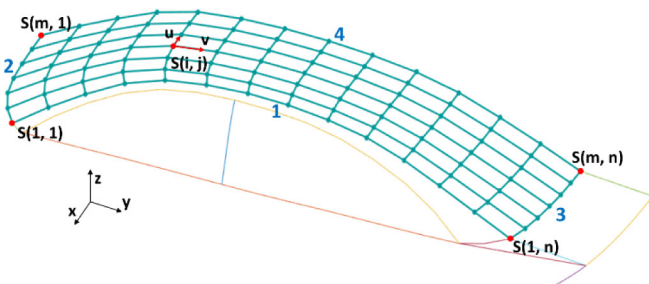


Figure 4. Notional representation of roof surface

The roof surface in Figure 4 is described in two parameters, u and v . These are functions of x, y , and z and eventually, functions of the parameter t . There are m and n points along u and v axes, respectively. Therefore, S_{ij} that is the point of x, y , and z of i -th along the u axis and j -th along the v axis on the surface S is represented as follows.

$$\vec{S}_{ij} = \begin{bmatrix} Sx_{ij} \\ Sy_{ij} \\ Sz_{ij} \end{bmatrix} = \begin{bmatrix} Sx(u_{ij}, v_{ij}) \\ Sy(u_{ij}, v_{ij}) \\ Sz(u_{ij}, v_{ij}) \end{bmatrix} = \begin{bmatrix} Sx(u_{ij}(x(t), y(t), z(t)), v_{ij}(x(t), y(t), z(t))) \\ Sy(u_{ij}(x(t), y(t), z(t)), v_{ij}(x(t), y(t), z(t))) \\ Sz(u_{ij}(x(t), y(t), z(t)), v_{ij}(x(t), y(t), z(t))) \end{bmatrix} \quad (7)$$

Where $i=1, 2, 3, \dots, m$ and $j=1, 2, 3, \dots, n$. Note that S is a 3rd-order tensor ($S \in \mathbb{R}^{3 \times m \times n}$) whereas Sx, Sy , and Sz are 2nd-order tensors ($Sx, Sy, Sz \in \mathbb{R}^{m \times n}$) and u and v are 1st-order tensors ($u, v \in \mathbb{R}^3$). In addition, $S_{a-b,c-d}$ represents the set of the points of x, y , and z : $S_{a-b,c-d} = \{S_{ij} | i=a, a+1, \dots, b-1, b; j=c, c+1, \dots, d-1, d\}$, where a, b, c , and d are positive integers. The subsequent mathematical backgrounds are described using these terminology conventions. First, a basis line from which points are derived is chosen. In this example, Line 1 is the basis line. This line is what we have obtained so far from Figure 2. Second, two path lines, along which the surface construction tasks will match, are chosen. Line 2 and 3 are the selected path lines. Once these basis and path lines are chosen, the surface begins to be constructed by repeating the two-layered arithmetic calculations. In summary, starting from the basis line, calculations clone lines one-by-one along both path lines until reaching the end line (Line 4 in Figure 4) to finish constructing the roof surface.

In this situation, we know that $S_{1,1-n}$ is the basis line (Line 1 in Figure 4). In the same way, $S_{1-m,1}$ is the first path line (Line 2), and $S_{1-m,n}$ is Line 3. Using this, at each calculation cycle, the following procedures arise. As we are currently starting from the first basis line, first, new boundary points are obtained at both ends along the v axis: $S_{1-m,1}$ and $S_{1-m,n}$. These correspond to Line 2 and Line 3, respectively in Figure 4. Then, the differences (or offsets), defined as d , for each point are calculated:

$$d_k = S_{i,k} - S_{i-1,k} \quad (8)$$

Where $k=1, 2, \dots, n$. Therefore, $d_1 = S_{i,1} - S_{i-1,1}$ and $d_n = S_{i,n} - S_{i-1,n}$. Finally, the difference of these differences is calculated by $\text{diff} = d_n - d_1$ and evenly distributed throughout the inner points, $S_{1-m,2-n-1}$, by equation 9.

$$S_{ij} = S_{i-1,j} + d_1 + \frac{j-1}{n-1} \times \text{diff} \quad (9)$$

The calculations in the z component are illustrated in Figure 5 for better understanding. For clarification, magnitude is exaggerated.