Soft and Hard Constrained Parametric Generative Schemes for Encoding and Synthesizing Airfoils

Hairun Xie*, Jing Wang† and Miao Zhang‡
Shanghai Aircraft Design and Research Institute, Shanghai, China, 200436

Traditional airfoil parametric technique has significant limitation in modern aerodynamic optimization design. There is a strong demand for developing a parametric method with good intuitiveness, flexibility and representative accuracy. In this paper, two parametric generative schemes based on deep learning methods are proposed to represent the complicate design space under specific constraints. 1. Soft-constrained scheme: The CVAE-based model trains geometric constraints as part of the network and can provide constrained airfoil synthesis; 2. Hard-constrained scheme: The VAE-based model serves to generate diverse airfoils, while an FFD-based technique projects the generated airfoils to the final airfoils satisfying the given constraints. The statistical results show that the reconstructed airfoils are accurate and smooth without extra filters. The soft constrained scheme tend to synthesize and explore airfoils efficiently and effectively, concentrating to the reference airfoil in both geometry space and objective space. The constraints will loose for a little bit because the inherent property of the model. The hard constrained scheme tend to generate and explore airfoils in a wider range for both geometry space and objective space, and the distribution in objective space is closer to normal distribution. The synthesized airfoils through this scheme strictly conform with constraints, though the projection may produce some odd airfoil shapes.

I. Introduction

As the requirements for economy and environmental friendliness of large commercial airliners become increasingly stringent, the aerodynamic design of supercritical wings faces new challenges. To address these challenges, the geometric parametric technique should be developed first to meet the increasing demand in engineering design. A small number of variables is preferred to represent a large enough design space, containing optimum aerodynamic shapes for a wide range of design conditions and constraints[1]. Besides, intuitive geometric interpretation is another urgent need for designers in order to build the design logic. However, the existing popular parametric approaches usually takes a trade-off between design freedom and intuitiveness[2-4]. For example, PARSEC[5] employs explicit mathematical functions for 2D curve
definition for airfoil design. In this way, the definitions are closely tied to geometric features with good intuitiveness. However, relative little flexibility and poor representative accuracy limits its application. NURBS method expresses complex geometries by using control points and related weights as design variables[6]. Its representation is naturally smooth and can avoid suffering noise and “bump”[7], making local modification easy to implement. However, it is difficult to coordinate the complex relationships that need to be satisfied between multiple control points for specific constraints[8], such as designing under a given thickness. Free Form Deformation (FFD) faces a similar problem, with a wide range of generative capabilities but poor correlation with geometric features[9]. Therefore, there is a strong demand for developing an parametric method with good intuitiveness, flexibility and representative accuracy.

Fig. 1 Classification of popular parametric approaches.

In recent years, machine learning has been applied in a wide range of fields, such as the computer vision, natural language process and recommendation algorithm. In particular, data-driven generative algorithms had been widely used in the engineering field, which were capable to automatically learn complex and comprehensive representations from samples and generate new samples. Generative Adversarial Network (GAN)[10] is one of the most popular generative deep learning algorithms for unsupervised learning in complex distribution. It contains at least two modules: The Generator and the Discriminator. The Generator generates fake samples of data and tries to fool the Discriminator. The Discriminator, on the other hand, tries to distinguish between the real and fake samples. The Generator and the Discriminator can both benefit from the minimax game. Variational AutoEncoder (VAE)[11] is another method to conduct a generative network, which includes two modules: the Encoder and the Decoder. The Encoder encodes input data to a latent vector space following specific probabilistic distribution. While the Decoder decodes latent vector to the desired output data. The training process balances the reconstruction loss and divergence from the distribution, which correspond to the accuracy and generalization of the generative model, respectively. Compared with GAN, VAE has better correspondence between latent vector and data, “pixel-level” consistency between input and output data. Besides, VAE is relatively easy to implement, and more stable in the training process. These characteristics of VAE are very much in line with the requirements for improving airfoil parametric methods. In addition to the basic generative models, the conditional generative models, such as Conditional Generative Adversarial Network (CGAN)[12] and Conditional Variational AutoEncoder (CVAE)[13], are more useful when processing data under conditional constraints.

In the present research, two parametric generative schemes are proposed to represent the complicate design space
under specific constraints. **1. Soft-constrained scheme:** The CVAE-based model trains geometric constraints as part of the network and can provide constrained airfoil synthesis. Because the error of the model can only be reduced but not eliminated, there is a certain error between the constraint and the generated target, so it is called soft constraint scheme; **2. Hard-constrained scheme:** The VAE-based model serves to generate diverse airfoils, while an FFD-based technique projects the generated airfoils to the final airfoils satisfying the given constraints. In this scheme the airfoil is strictly conform with constraints, although the projection may produce some odd airfoil shapes.

Experiments demonstrate that both schemes have the ability to precisely parameterize either existing or novel airfoil shape, as well as generating new practical smooth airfoils even without any smoothing post-process. With further research on the aerodynamic properties, the hard-constrained scheme has a larger coverage of aerodynamic parameters than soft-constrained scheme when performing perturbations based on an original airfoil.

**II. Related works**

To better articulate the focus and idea of current work, previous researches on airfoil parameterization especially with deep learning frames are reviewed in this section.

GAN is a popular technique in airfoil shape parameterization. Chen et al. [14,15] introduced BézierGAN to generate Bézier curve control points by GAN, then used the control points to formulate the airfoils. The pipeline guaranteed smooth airfoil shape and reduction of feature domain. However, the Bézier curves restricted the diversity and failed to connect geometry feature explicitly. Du et al. [16,17] proposed a similar method BSplineGAN to generate airfoils and applied it into a fast interactive aerodynamic optimization framework. Based on that, a Webfoil toolbox was developed to perform interactive airfoil aerodynamic optimization on any modern computing device. Achour et al.[18] used CGAN to generate airfoil under specific aerodynamic characteristics such as lift-to-drag ratio (L/D) and structural requirements (shape area). This sparked the idea of incorporating optimization objectives into deep learning models and conducting optimization through the training process. Li et al.[19] proposed deep CGAN to sample airfoils and detect the geometric abnormality quickly without using expensive computational fluid dynamic models, and embedded in a surrogate-based aerodynamic optimization framework to perform airfoil aerodynamic optimization.

VAE is another popular generative algorithm to parametric airfoils. Wang et al.[20] built a VAE-GAN model combining VAE and GAN, which can encode and synthesize airfoils by interpolating/extrapolating learned features. Wang et al.[21] proposed a CVAE-GAN based inverse design approach to generate airfoils for given pressure distribution, while the smoothness measurement can prove the authenticity and accuracy of airfoil. Yonekura and Suzuki[22] used CVAE to analyze the relationship between aerodynamic performance and the airfoil shape, and the airfoil can be generated for desired aerodynamics performance. However, some of the airfoils produced by this strategy may not be realistic and smooth.

Existing works demonstrate the power of GAN and VAE, but do not provide an in-depth discussion of the geometric
constraints in synthesizing airfoils, which is very important in airfoil design.

III. Methodology

A. Dataset

Dataset is crucial for both model training and validation, determining the reliability and applicability of the research. According to different purposes, three datasets are used here:

1. UIUC Dataset (Training)

The training dataset is chosen from the (University of Illinois Urbana-Champaign) UIUC Coordinates Database, which contains more than 1500 airfoils. There is a rich diversity of airfoil types in the dataset, which can be applied to wind turbine blades, compressor blades, aircraft wings, etc. The airfoils of the aircraft wing include low-speed airfoil, medium-speed airfoil, supercritical airfoil, etc. As a training set, UIUC Dataset (denoted by Training in this paper) has been widely used in many studies[18][20].

Fig. 2 Typical airfoils in UIUC dataset.

2. Random NACA 4 Digits Dataset (Test1)

NACA Four-digits airfoil is a classical and simple airfoil parameterization method, which uses only three parameters to define an airfoil: the maximum camber, the relative position of maximum camber and the maximum thickness. For example, NACA 2412, which designate the camber 2% of chord, position of the maximum camber 40% of chord and maximum thickness 12% of the chord. According to this definition, the Random NACA 4 digits Dataset is randomly generated by using more refined values and includes 1500 airfoils. Latin Hypercube Sampling is used to sampling the three parameters in the range of existing NACA airfoils in UIUC dataset. Since the UIUC dataset contains many NACA airfoils, the Random NACA 4 digits Dataset can be regard as an interpolation of the UIUC dataset and will be used as the interpolation test set (denoted by Test1 in this paper).

3. Supercritical Airfoil Dataset (Test2)

In the previous research[21], a supercritical airfoil dataset had been created starting from a new airfoil design. This dataset keeps diversity from the geometric feature to aerodynamic performance, and applies several constraints, such as the leading edge radius must not be smaller than 0.007 and the drag coefficient should not exceed 0.1 to avoid impractical
Fig. 3 Typical airfoils in random NACA 4 digits dataset.

Because none of the airfoils in this dataset is included in UIUC dataset, it can be regard as an extrapolation test set (denoted by Test2 in this paper).

Fig. 4 Typical airfoils in supercritical airfoil dataset.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The t-SNE plot of all three datasets is shown in Fig. 5. The interrelationships and boundaries of the three datasets are very clear. The Training is widely distributed, and the coverage of the Test1 overlaps with the Training, and the boundary between the Test2 and the other two datasets is very clear. The closest sample in Training to Test2 is NASA SC(2)-0710, which comes from a matrix of family-related supercritical airfoils with thicknesses from 2 to 18 percent and design lift coefficients from 0 to 1.0[23].

B. Pre-processing

Pre-processing includes rotation, scale, resampling and verification. In the first step, rotation/scale transforms the leading edge and trailing edge points to [0,0] and [1,0] in Cartesian coordinates. In the second step, the different airfoil coordinate formats due to the diverse sources of the datasets require resampling to create homogenous representation of all airfoils. Cosine distribution of x coordinates given in Eq. (1).

\[
\theta_i = \frac{\pi(i-1)}{N}
\]

\[
x_i = 1 - \cos (\theta_i)
\] (1)

Where N is set to 201 to ensure the 100 segments on both upper surface and lower surface. The corresponding y coordinates are interpolated by a cubic spline curve. Since all airfoils share the same x coordinates, the airfoil geometry can be represented by the y coordinates, which are fed to the model. In the third step, geometry features are extracted,
such as maximum thickness, maximum camber, leading edge radius, etc. In the last step, normalization is applied to $y$ coordinates and geometry features, respectively. Normalization of $y$ coordinates is based on normalization coefficients obtained from training set, scaling the training set to $[-1,1]$ range. Normalization of geometry features is a little different, with each feature normalized by the corresponding normalization coefficients independently to $[-1,1]$ range.

Fig. 6 demonstrates the max camber vs. max thickness distribution of the datasets, the distribution of the Training set is close to the normal distribution except that it contains a lot of zero camber samples. The Test1 set is widely distributed while the Test2 set is very concentrated with the same max thickness.
C. Learning models

1. VAE

![Fig. 7 The base VAE architecture.](image)

The base VAE architecture shown in Fig. 7 is the foundation of generative schemes. The airfoil coordinates $y$ is taken as the input, and the synthesized airfoil coordinates $y'$ is taken as the output. The encoder and decoder are constructed by MLP with similar structure. The encoder maps $y$ to learned latent distribution $z$, and the decoder generate $y'$ from the latent vector $z$. The encoder and decoder can be expressed as:

$$ z \sim \text{Enc}(y) = q(z \mid y), \quad y' \sim \text{Dec}(z) = p(y' \mid z) \quad (2) $$

where $q(z)$ is the learned distribution and $p(z)$ is the prior distribution. The loss function of VAE is given by:

$$ L_{VAE} = \alpha_1 L_{mse} + \alpha_2 L_{kld} \quad (3) $$

where $L_{mse} = \|y - y'\|^2_2$ and Kullback-Leibler divergence $L_{kld} = D_{kl}(q(z \mid y) || p(z))$. $L_{mse}$ measures the error between original and synthesized airfoil, minimizing it leads to higher reconstruction accuracy. $L_{kld}$ measures the similarity between the learned and prior distribution, the smaller $L_{kld}$ results in more effective airfoil generation based on prior distribution. The generative capability introduces errors that affect the reconstruction accuracy during training, while the improvement of reconstruction accuracy also weakens the generative capability. $\alpha_1$ and $\alpha_2$ are hyperparameters used to balance them.

2. Soft constrained scheme

The soft constrained scheme is developed based on CVAE. In CVAE, the Encoder maps $y$ to $z$ with distribution $q(z \mid y, c)$ under specific given conditions. The Decoder generates $y'$ with latent vector $z$ under condition $c$. The loss function is the same as VAE. It is obvious seen that the geometry constraint $c$ exactly matches the original sample $y$, but there is a certain error with the generated airfoil $y'$, which can be only reduced but not eliminated with the training process.
3. Hard constrained scheme

The hard-constrained scheme combines VAE with a hard projection process. The VAE generates diverse airfoils $y'$, and hard projection makes airfoil $y''$ satisfying the desired geometric constraints $c$. The hard projection is realized through the common parameterization method FFD. 2D Bezier-based FFD is defined in terms of bivariate Bernstein polynomial. Displacement $\Delta x$ of any point $x(s, t)$ in the control box can be expressed as:

$$x(s, t) + \Delta x(s, t) = \sum_i \sum_j B_{i-1}^{l-1}(s) B_{m-1}^{j-1}(t) \left[ P_{i,j} + \Delta P_{i,j} \right]$$  \hspace{1cm} (4)

where $P_{i,j}$ and $\Delta P_{i,j}$ are the original coordinates and displacements of the vertices of the control box, respectively. The Eq. (4) can be rewritten into the matrix form:

$$\Delta x = B \cdot \Delta P$$  \hspace{1cm} (5)

In the concurrent application, $\Delta x$ is obtained from the geometry constraints, $B$ remains constant when building up
the control box, and $\Delta P$ is the only unknown variable. Fig. 10 illustrates a simple example to project the thickness distribution to the given constraints. The maximum thickness and maximum thickness position are the given constraints, which can be represented by the peak point of thickness distribution curve. The control box includes 6 vertices. Thus, $\Delta x$ includes 2 variables ($[dx, dy]$ of peak point), $\Delta P$ includes 12 unknowns ($[dx, dy]$ of 6 control vertices, Eq.(5) is underdetermined with infinite solution set. In order to obtain one solution with better curve characteristics, the degree of freedom of the control box is partly released. In this case, vertex 2/5 share the same $dx$, vertex 4/5/6 share the same $dy$, while other degrees of $\Delta P$ are freeze. Then $\Delta P$ can be solved for 2 unknowns, and deformation of any other points on the curve can be derived from the equation.

Combining VAE and FFD projection, the hard-constrained scheme gains generative capability while strictly satisfying the geometry constraints.

### IV. Numerical experiment

#### A. Hyperparameters

Basic hyperparameters of the predictive model is tuned based on the VAE model, with only the training set feeding in, thus the Test1 and Test2 is completely unknown to the model.

The encoder contains 3 hidden layers with 256, 128 and 64 neurons, respectively. The decoder contains 3 hidden layers with 64, 128, 256 neurons, respectively. Dimension of the latent vector $z$ is set to be 10. Five geometry constraints (max thickness position, max thickness, trailing edge thickness, max camber position and max camber) are taken as the condition $c$.

The hyperbolic tangent (Tanh) activation function is employed after all layers except the output layer of both the
encoder and decoder. Adam optimizer is used to optimize the loss function. The model is trained for 5000 epochs, the learning rate is initially set to be 5.0e-4 and decays to 5.0e-5 after 2500 epochs.

In addition, two crucial hyperparameters, prior distribution type and $\alpha_1$ in loss function, will be discussed in detail. These two factors will be discussed together because both of them can lead to collapse or divergence of the latent space vectors. VAE using the Gaussian variational prior distribution (denoted by N-VAE in this section), shows collapse in low dimensions and ‘soap-bubble-effect’ in high dimensions. To solve these issues, VAE with a hyperspherical latent space\cite{24} (denoted by S-VAE in this section) is developed. Besides, $\alpha_1$ can also help prevent collapse.

\begin{figure}[h]
\centering
\begin{subfigure}{0.45\textwidth}
\includegraphics[width=\textwidth]{l_mse.png}
\caption{$L_{mse}$}
\end{subfigure}
\begin{subfigure}{0.45\textwidth}
\includegraphics[width=\textwidth]{l_kld.png}
\caption{$L_{kld}$}
\end{subfigure}
\caption{Influence of distribution type and $\alpha_1$.}
\end{figure}

Fig. 11 illustrates the influence of distribution type and $\alpha_1$ on the loss function of VAE model, indicating the competitive relationship between $L_{mse}$ and $L_{kld}$. As $\alpha_1$ increases from $10^2$ to $10^6$, $L_{mse}$ of both N-VAE and S-VAE decreases and remains almost constant after $\alpha_1$ reaching $10^6$. $L_{mse}$ of the N-VAE is smaller than S-VAE for all $\alpha_1$, indicating higher reconstruction accuracy of N-VAE. On the other side, $L_{kld}$ of the N-VAE keeps increasing as $\alpha_1$ increases while $L_{kld}$ of the S-VAE remains unchanged after $10^5$. If $L_{kld}$ is greater than 30.0, there are a lot of bad airfoils when sampling from the prior distribution. Therefore, S-VAE succeeds in preventing the divergence of $L_{kld}$. Overall, N-VAE performs better in reconstruction accuracy, and SVAE is able to maintain strong generative ability even when $\alpha_1$ is quite large. When comparing the distribution type and $\alpha_1$, N-VAE at $\alpha_1 = 10^5$ appears to be the best choice to balance the accuracy and generative capability.

Since VAE is the basic building block of the soft and hard schemes, local changes to the neural network have little impact on the hyperparameter selection. Each scheme takes the same hyperparameters and performs well.
### Table 1  Results of reconstruction errors

| Datasets  | Soft-constrained | Hard-constrained |
|-----------|------------------|------------------|
|           | MSE   | MAE   | MSE   | MAE   |
| Training  | 3.60E-07 | 4.51E-04 | 5.36E-07 | 5.46E-04 |
| Test1     | 5.53E-07 | 5.22E-04 | 8.42E-07 | 6.35E-04 |
| Test2     | 2.50E-06 | 1.14E-03 | 2.13E-06 | 1.11E-03 |

Fig. 12  Comparisons of reconstructed airfoils.

**B. Airfoils Reconstruction**

Errors for the two generative schemes are given in Table 1. The soft-constrained scheme has smaller MSE and MAE on Training set and Test1 set, whereas the hard-constrained performs slightly better on Test2 set. Errors of the Test2 is greater than Test1, indicating that extrapolation is more challenging than interpolation. Original and reconstructed airfoils from all the datasets are shown in Fig. 12. When compared to the original airfoils, all of the reconstructed
airfoils are smooth and accurate.

The geometric features of the reconstructed airfoils through hard-constrained scheme are strictly identical to the original airfoils because of precise projection in hard-constrained scheme. However, there are certain errors in soft-constrained scheme. The error distributions of max thickness and max camber are shown in Fig. 13. It is seen that 2.5% error band covers all samples of all the datasets for the max thickness, and 98.5% samples for the max camber.

![Fig. 13 Error distribution of geometry features for soft-constrained scheme.](image)

C. Airfoils Synthesizing

In generative schemes, airfoils are encoded into latent space first, then the latent vectors are employed as variables to synthesize new airfoils, with no explicit relationship between the geometric characteristics and the latent space. The latent vector distribution of VAE, for example, is shown in Fig. 14. For each dimension, the Training set has been well-trained, with Test1 and Test2 clustered in specific regions. In particular, \( z_5 \) of Test2 is distributed in a very narrow band, looking back to Fig. 5 the thickness distribution of Test 2 is also in a thin band, this could be a possible mapping. It demonstrates that it is possible to control the geometric features to some extent according to the characteristics of latent vector distribution.

Airfoil synthesizing based on the latent vectors is shown in Fig. 15. The reference airfoil is generated from a reference latent vector \( z_{ref} = 0 \in \mathbb{R}^{10} \). The synthesizing airfoils are generated with \( z_1 \) to \( z_6 \) varying from -4.0 to 4.0 and other vectors equaling the reference vector. It is seen that the deformation caused by \( z_1 \) is mainly located near the trailing edge, while \( z_2 \) affects both the leading and trailing edges. \( z_4 \) primarily controls the max thickness position. \( z_3 \) seems to play an important role in smoothness, which is a higher order representation of airfoil. \( z_5 \) is clearly related to the max thickness, whereas \( z_6 \) is critical for adjusting the max camber position. The preceding comparisons show that different latent vectors affect airfoil geometry at various scales and in different areas at various orders. The variation
Fig. 14  Latent vector distribution of VAE.

The latent space of $z$ covers almost all samples, representing that the range of the geometric features variation of the synthesized airfoils can match the whole datasets, achieving the completeness of geometry space.

Fig. 15  Airfoil synthesizing of VAE.
In the case of optimization design based on existing airfoil, the soft and hard constrained schemes can be used to provide geometry parametric variables to generate many new airfoils while satisfying the constraints.

Fig. 16 presents 500 airfoils synthesized by two schemes based on NASA SC(2)-0710 with 5 geometry constraints (max thickness position, max thickness, trailing edge thickness, max camber position and max camber). Random variations in the range [-1, 1], through Latin Hypercube Sampling (LHS), are superimposed on the latent vector of the reference airfoil to synthesize new airfoils from these latent vectors. It is seen that the variance of the hard constrained scheme is greater than that of the soft constrained scheme, indicating that it covers more geometry space. The airfoils generated by the soft constrained scheme are all smooth and reasonable, whereas some of the airfoils generated by the hard constrained scheme are too odd to be removed.

In the traditional parametric methods, the airfoil generation is tightly limited by so many geometry constraints, making it difficult to produce variance of airfoils. For example, NACA 4 digits airfoil is over-constrained and no airfoil can be generated. When it comes to CST/FFD/NURBS, it is even impossible to apply constraints before generation, and probably no airfoil can be left after the post generation screening. Therefore the capability to produce as many as desired samples under strict constraints of the proposed schemes is superior than existing parametric methods.

In addition to the design space, the diversity of the objective parameters space is also concerned. An in-house code for quick aerodynamic performance evaluation based on the corrected full potential equation is used to give a preliminary check of the objective space distribution. Fig. 17 shows the distribution of moment coefficient and lift to drag ratio for synthesized airfoils. It can be seen that the samples from the soft scheme are more concentrated near the reference, and very few airfoils perform very high or low lift to drag ratio. In this way, the obtained distribution is biased, which may lead to blind spots in the objective space. For the hard scheme, the generated samples vary over a greater region, and the distribution near the reference is more uniform. This is consistent the geometries distribution. The hard scheme tends to explore extreme region, there’s some probability that it can produce some samples with very
In summary, the two schemes can be used in different application scenarios based on their objective space distribution characteristics. The soft scheme, with higher sampling efficiency, effectiveness, and clustered distribution near the reference airfoil in objective space, would be ideal for local optimization for an good airfoil. Once a local optimal is found, sampling can be restarted from the new one. The hard scheme, with wider distribution range, close to normal distribution in objective space, is suitable for global optimization to explore better extreme performance.

Fig. 17 Objective space distribution.

V. Conclusions

In present work, a data-driven deep learning approach is used for airfoil geometry parametric technique. VAE based generative schemes with soft and hard constraints are developed. The conclusions are as follows:

1) Through detailed study of VAE characteristics, the airfoil parametric schemes are implemented with capability of encoding and synthesizing airfoils. By encoding the exiting airfoils, the geometry features are reduced into reduced order latent space, and can be reconstructed with high accuracy and smoothness. The reconstruction accuracy outperforms GAN in [20] more than one order of magnitude. And the airfoil is smooth enough for calculation without any smooth filters. The “pixel-level” consistency of VAE is the key factor to achieve this.

2) The latent space distribution of datasets proves the completeness of generated samples in geometry space. The latent vector of well-trained model is able to reflect the complete characteristics of an airfoil, and can be treated as the variable for airfoil parameterization. The combination of these characteristics can represent both the interpolation and the extrapolation of known samples. The variance of the latent vector reflects the variant
geometry features of synthesized airfoil based on reference.

3) The soft constrained scheme tend to synthesize and explore airfoils efficiently and effectively, concentrating to the reference airfoil in both geometry space and objective space. The constraints will loose for a little bit because the inherent property of the model.

4) The hard constrained scheme tend to generate and explore airfoils in a wider range for both geometry space and objective space, and the distribution in objective space is closer to normal distribution. The synthesized airfoils through this scheme strictly conform with constraints, though the projection may produce some odd airfoil shapes.

References

[1] Kulfan, B. M., “Universal Parametric Geometry Representation Method,” *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158. https://doi.org/10.2514/1.29958 URL https://arc.aiaa.org/doi/10.2514/1.29958

[2] Liao, Y. P., Liu, L., and Long, T., “Investigation of Various Parametric Geometry Representation Methods for Airfoils,” *Applied Mechanics and Materials*, Vol. 110-116, 2011, pp. 3040–3046. https://doi.org/10.4028/www.scientific.net/AMM.110-116.3040 URL https://www.scientific.net/AMM.110-116.3040

[3] Derksen, R., and Rogalsky, T., “Bezier-PARSEC: An optimized aerofoil parameterization for design,” *Advances in Engineering Software*, Vol. 41, No. 7-8, 2010, pp. 923–930. https://doi.org/10.1016/j.advengsoft.2010.05.002 URL https://linkinghub.elsevier.com/retrieve/pii/S0965997810000529

[4] Zhu, F., Qin, N., Burnaev, E., Bernstein, A., and Chernova, S., “Comparison of Three Geometric Parameterisation Methods and Their Effect on Aerodynamic Optimisation,” 2011, p. 5.

[5] Sobieczky, H., “Parametric Airfoils and Wings,” *Recent Development of Aerodynamic Design Methodologies*, Vol. 65, edited by E. H. Hirschel, K. Fujii, W. Haase, B. van Leer, M. A. Leschziner, M. Pandolfi, A. Rizzi, B. Roux, K. Fujii, and G. S. Dulikravich, Vieweg+Teubner Verlag, Wiesbaden, 1999, pp. 71–87. https://doi.org/10.1007/978-3-322-89952-1_4 URL http://link.springer.com/10.1007/978-3-322-89952-1_4 series Title: Notes on Numerical Fluid Mechanics (NNFM).

[6] Liang, Y., Cheng, X., Li, Z., and Xiang, J., “Multi-objective robust airfoil optimization based on non-uniform rational B-spline (NURBS) representation,” *Science China Technological Sciences*, Vol. 53, No. 10, 2010, pp. 2708–2717. https://doi.org/10.1007/s11431-010-4075-4 URL http://link.springer.com/10.1007/s11431-010-4075-4

[7] Lepine, J., Trepanier, J.-Y., and Pepin, F., “Wing aerodynamic design using an optimized NURBS geometrical representation,” *38th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Reno,NV,U.S.A., 2000. https://doi.org/10.2514/6.2000-669 URL https://arc.aiaa.org/doi/10.2514/6.2000-669

[8] Painchaud-Ouellet, S., Tribes, C., Trépanier, J.-Y., and Pelletier, D., “Airfoil Shape Optimization Using a Nonuniform
[19] Li, J., Zhang, M., Martins, J. R. R. A., and Shu, C., “Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering,” *AIAA Journal*, Vol. 58, No. 10, 2020, pp. 4243–4259. https://doi.org/10.2514/1.J059254, URL https://arc.aiaa.org/doi/10.2514/1.J059254.

[20] Wang, Y., Shimada, K., and Farimani, A. B., “Airfoil GAN: Encoding and Synthesizing Airfoils for Aerodynamic-aware Shape Optimization,” *arXiv:2101.04757 [cs]*, 2021. URL http://arxiv.org/abs/2101.04757, arXiv: 2101.04757.

[21] Wang, J., Li, R., He, C., Chen, H., Cheng, R., Zhai, C., and Zhang, M., “An inverse design method for supercritical airfoil based on conditional generative models,” *Chinese Journal of Aeronautics*, 2021, p. S1000936121000662. https://doi.org/10.1016/j.cja.2021.03.006, URL https://linkinghub.elsevier.com/retrieve/pii/S1000936121000662.

[22] Yonekura, K., Miyamoto, N., and Suzuki, K., “Inverse airfoil design method for generating varieties of smooth airfoils using conditional WGAN-gp,” *arXiv:2110.00212 [cs]*, 2021. URL http://arxiv.org/abs/2110.00212, arXiv: 2110.00212.

[23] Harris, C. D., “NASA supercritical airfoils: A matrix of family-related airfoils,” Mar. 1990. URL https://ntrs.nasa.gov/citations/19900007394, nTRS Author Affiliations: NASA Langley Research Center NTRS Report/Patent Number: NASA-TP-2969 NTRS Document ID: 19900007394 NTRS Research Center: Legacy CDMS (CDMS).

[24] Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M., “Hyperspherical Variational Auto-Encoders,” *arXiv:1804.00891 [cs, stat]*, 2018. URL http://arxiv.org/abs/1804.00891, arXiv: 1804.00891.