Aerospace Science and Technology 166 (2025) 110548

Contents lists available at ScienceDirect

Aerospace Science and Technology

journal homepage: www.elsevier.com/locate/aescte

# CST-LLM: Enhancing airfoil parameterization method with large language model

Kefeng Zheng [a], Yiheng Wang [b], Fei Liu [a,], Qingfu Zhang [a,*], Wenping Song [b]

[a] Department of Computer Science, City University of Hong Kong, Hong Kong, China
[b] Institute of Aerodynamic and Multidisciplinary Design Optimization, School of Aeronautics, Northwestern Polytechnical University, Xi'an, China

ARTICLE INFO

Communicated by Kivanc Ekici

*Keywords:*
Large language model
Airfoil parameterization method
Algorithm design
Airfoil design optimization
Symbolic regression

ABSTRACT

The airfoil parameterization method is pivotal for airfoil design optimization, as it transforms numerous discrete coordinate points into a concise and manageable set of parameters. The primary goal of the parameterization method is to achieve a more precise representation and control of airfoil geometry with fewer parameters. This paper proposes a symbolic regression method based on the pre-trained Large Language Model (LLM) for enhancing the performance of the airfoil parameterization method. The proposed method utilizes LLM to generate and evolve feature transformation functions for modifying the Class/Shape function Transformation (CST) method. By offering simple mathematical insights to the LLM through natural language, feature transformation functions are effectively designed and the CST-LLM parameterization method is proposed by modifying the CST parameterization method using designed function. In geometry recovery tests conducted on open source airfoil libraries, the CST-LLM parameterization method consistently outperforms other state-of-the-art parameterization methods, offering a more complete design space for airfoil design optimization by accurately representing a wider range of airfoil geometries. In design optimization of RAE 5214 airfoil, the CST-LLM parameterization method consistently achieved better results compared with the CST parameterization method. These improvements not only demonstrate the efficacy of the proposed LLM-based symbolic regression method, but also underscore the wide application potential of LLM in scientific research.

## 1. Introduction

In airfoil design optimization, the parameterization method is applied for representing numerous discrete coordinate points with a limited number of parameters to achieve precise control and design of airfoil geometries [1–5]. Airfoil parameterization methods directly determine the design space which refers to the range and collection of airfoil geometries available for selection during design optimization. A well-performed parameterization method ensures that the design space encompasses a diverse array of feasible airfoil geometries, providing designers with a rich set of options [5–7]. Besides, airfoil parameterization method has a profound impact on the mathematical characteristics of the optimization problem [5,8,9] and affect the convergence speed as well as the final results.

These considerations motivate the development of advanced parameterization methods for airfoil design optimization. An ideal parameterization method should simultaneously achieve precise geometric representation with fewer control parameters. In recent years, a variety of airfoil parameterization methods have been proposed. These methods can be roughly classified into deformative methods and constructive methods [2,3]. The deformative methods take an existing airfoil and deform it to create new airfoils. For example, the Hicks-Henne method [10] takes a base airfoil and adds a linear combination of single signed sine functions to deform its upper and lower surfaces to create new airfoils. The Free Form Deformation (FFD) method [11–13] creates smooth and continuous volume transformations of airfoil geometry based on the movement of a series of control points.

Different from the deformative parameterization methods, the constructive parameterization methods represent the airfoil geometries based on a series of specified parameters. For example, the Parametric Section (PARSEC) method [14,15] represents the airfoil's upper and lower surface using $6^{th}$ order polynomials derived from real geometric

* Corresponding author.
*E-mail address:* qingfu.zhang@cityu.edu.hk (Q. Zhang).

https://doi.org/10.1016/j.ast.2025.110548
Received 11 January 2025; Received in revised form 27 April 2025; Accepted 25 June 2025
Available online 4 July 2025

properties. The B-spline method [16,17] represents a versatile class of piecewise polynomial curves based on control points. The Class/Shape function Transformation (CST) method [18,19] employs class and shape functions to represent the geometry of the airfoil. For a specific airfoil, the deformative methods have a stronger fitting ability compared with the constructive methods [20]. However, as the airfoil library scales up, the constructive method demonstrates superior parametric efficiency, maintaining accurate representation while requiring fewer parameters [21].

Extensive researches have systematically evaluated and compared the performance of various parameterization methods in aerodynamic design optimization [2,3,5–7,22,23]. Masters et al. [2,3] tested six parameterization methods including CST, B-splines, Hicks-Henne, RBF (Radial Basis Functions), Bézier surfaces, SVD and PARSEC method in geometry recovery test on over 1,000 airfoils. The CST parameterization method and its improved version, CST-LEM (Leading-Edge Modification) method [24], demonstrated excellent geometry representation capabilities in these tests. Guan et al. [22] compared the performance of the CST parameterization method with the Hick-Henne, B-spline and PARSEC method in the geometry recovery tests of different airfoils. They concluded that the CST parameterization method could maintain high accuracy and smoothness of geometry with fewer control parameters compared with other methods. Nadarajah et al. [6] compared different parameterization methods including CST, B-spline and PARSEC in drag minimization and inverse design of airfoils. Results show that the CST parameterization method achieves better results in design optimization with fewer design variables (less than 22) compared with the B-spline and other methods. Sripawuadkul et al. [7] conducted a comparison of the Hicks-Henne, B-spline, PARSEC and CST parameterization methods. They assigned scores for each method based on five criteria including parsimony, completeness, orthogonality, flawlessness and intuitiveness. The CST parameterization method received the highest rating using these criteria, followed by PARSEC, B-splines and Hicks-Henne. These studies highlight the exceptional advantages of the CST parameterization method, which make the CST parameterization method one of the most widely used parameterization methods in the field of aerodynamic design optimization.

Given the wide variety of airfoil types, each exhibiting distinct geometric characteristics, researches [2,3] have shown that existing airfoil parameterization methods require a minimum of 25 parameters to accurately represent over 80% of airfoils in large airfoil libraries within the wind tunnel test tolerance. This requirement exceeds the optimal parameter count that is typically preferred for the design optimization of two-dimensional geometries. Besides, for CST parameterization method, increasing the number of control parameters will make the parameterization matrix ill-conditioned [22,25,26]. This phenomenon not only affects the representation precision of airfoil geometry but also makes the parameterization process ill-posed. According to previous research [22,25], it is recommended to use Bernstein polynomials of the order between 4 to 10 (10 to 22 control parameters) in the CST parameterization method for representing airfoil geometries. However, the inability to accurately represent airfoil geometries with a limited number of parameters raises concerns about the completeness of the design space for optimization, potentially affecting the efficiency and effectiveness of airfoil design optimization. These limitations underscore the urgent need for more effective parameterization methods with enhanced geometric representation capabilities. However, the design and modification of airfoil parameterization methods require a comprehensive understanding of both mathematics and aerodynamics. This presents challenges for researchers and highlights the urgent need for more efficient methodologies to enhance the performance of airfoil parameterization method while requiring less expertise.

Recently, Large Language Models (LLM), which are trained on large datasets comprising publicly available source codes and natural language [27], have gained prominence in scientific discovery. The substantial size of LLM and the richness of their training data contribute to their increasing effectiveness in various research domains. Using LLM for function and algorithm design [28–30] have been successful applied in research fields including chemistry [31,32], biology [33] and mechanics [34,35]. The LLM offers a versatile and automated framework for scientific discoveries and significantly reduce the dependence on human intervention and specialized domain expertise [36–38]. Inspired by the Evolution of Heuristics (EoH) [29] used for algorithm design, in this paper, an LLM-based symbolic regression method using an evolutionary framework is proposed to modify the airfoil parameterization method. Comparisons with the Genetic Programming (GP)-based symbolic regression method are conducted. Results reveal that unlike GP-based symbolic regression method [39], the proposed LLM-based method does not require pre-specification of basic mathematical operators. This flexibility provides a free choice of mathematical operators during symbolic regression. Additionally, the incorporation of natural language dialogue as prompts allows designers to convey their requirements and preferences more conveniently than the cumbersome coding process required in the GP-based symbolic regression method [40–42]. Compared with using LLM directly for symbolic regression, integrating LLM into an evolutionary framework is more effective since individuals generated by LLM are precisely evaluated and ranked using fitness value calculated by the objective function. This process effectively eliminates unsuitable results produced by LLM during evolution, allowing for the efficient filtering of erroneous outputs, such as those arising from hallucinations [43].

In this paper, the CST parameterization method is used as the baseline parameterization method due to its board application and excellent performance in airfoil design optimization [2,3,22], with feature transformation functions $t(x)$ designed to modify the CST parameterization method by substituting the independent variable $x$. Datasets containing over 1,500 airfoils from UIUC airfoil libraries, including MH, Eppler, NACA, Clark, ONERA, RAE and other airfoil families, are used for training and testing. The objective is defined as the weighted sum of errors between the fitted airfoils and the original airfoils in the geometry recovery test. Prompts with different levels of mathematical insights are adopted for designing the feature transformation function and finally the LLM-enhanced CST parameterization method (CST-LLM) is introduced using the designed feature transformation function. In geometry recovery tests conducted on the testing set, the CST-LLM parameterization method consistently outperformed other state-of-the-art parameterization methods. Indicating that the CST-LLM parameterization method is able to provide a more complete design space for airfoil design optimization. For further validation, design optimizations of RAE 5214 airfoil are conducted using CST and CST-LLM parameterization methods and better results are achieved by the CST-LLM parameterization method. These results strongly demonstrate the effectiveness of not only the CST-LLM parameterization method, but also the proposed LLM-based symbolic regression method. The contributions of this paper are as follows.

1. Propose an LLM-based symbolic regression method to enhance the airfoil parameterization method.

2. Modify the CST parameterization method using feature transformation functions $t(x)$ designed by the proposed LLM-based symbolic regression method.

3. Explore strategies to integrate LLM into scientific research, provide a reference for the adaptation of LLM to address more complex scientific problems.

This paper is organized as follows. Section 2 provides a detailed introduction to the CST parameterization method. Section 3 introduces the framework of the proposed LLM-based symbolic regression method for enhancing the CST parameterization method. Section 4 presents numerical experiments aimed at designing feature transformation functions using the LLM-based symbolic regression method. In addition, the performance of the GP-based symbolic regression method is compared and discussed in Section 4 as well. Section 5 validates the effectiveness of

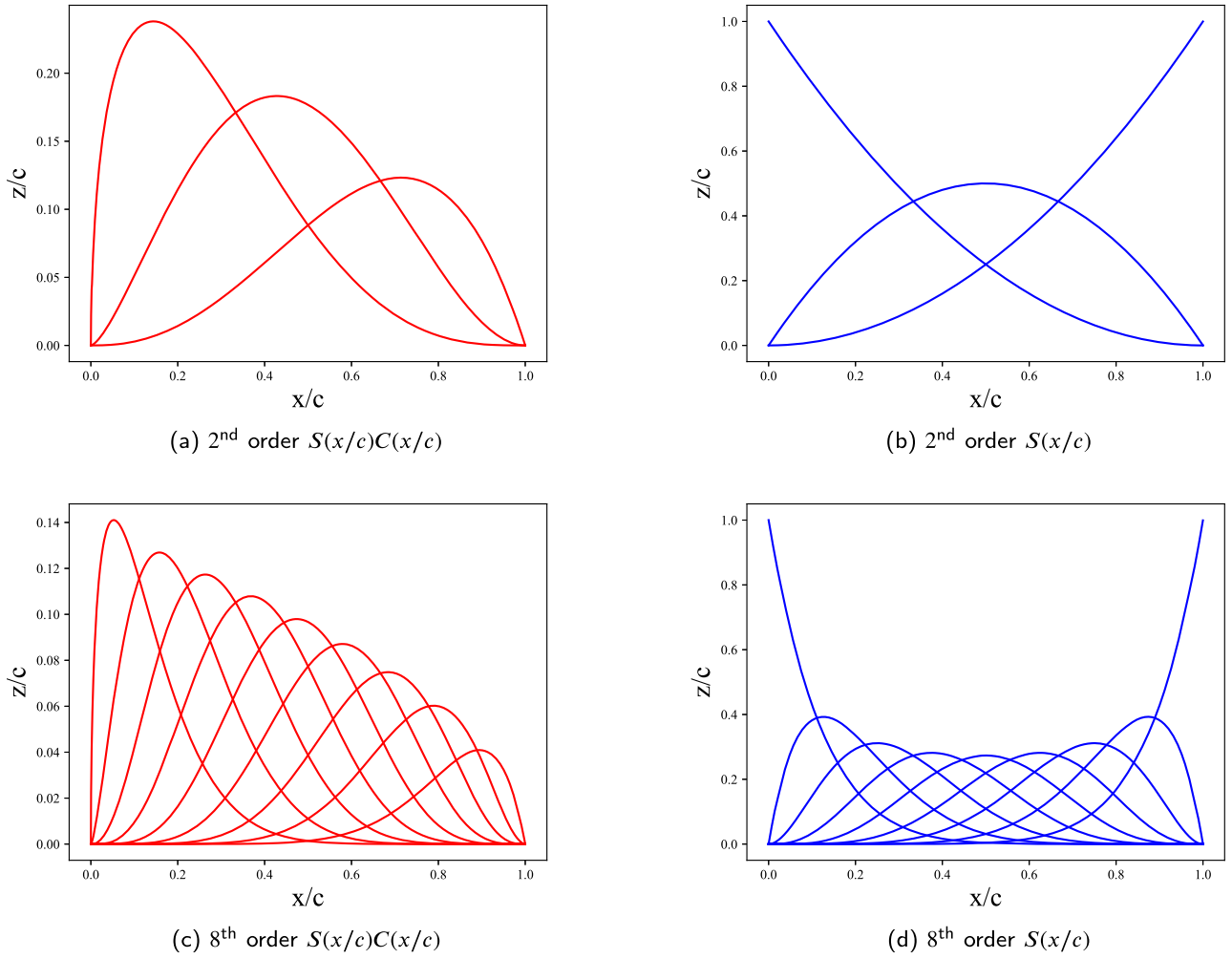



**Fig. 1.** $S(x/c)$ and $S(x/c)C(x/c)$ for different orders of the CST parameterization method ($N_1 = 0.5, N_2 = 1$).

the CST-LLM parameterization method in geometry recovery tests and airfoil design optimization. Finally, Section 6 presents the conclusion of this paper, discussing the limitations of the proposed method and potential directions for future research.

## 2. Overview of the Class/Shape function Transformation (CST) method

The Class/Shape function Transformation (CST) method [18,19,44] is a prominent parameterization method used in aerospace engineering to represent complex geometries, including airfoils, wings and fuselages [45]. This method is recognized for its high fidelity in capturing intricate geometric features using few parameters. Additionally, it demonstrates robust smoothing capabilities that enable the generation of smooth and continuous shapes. The mathematical formulation of the CST parameterization method for representing a 2D geometry, such as an airfoil, is delineated in Eq. (1):

$$z = C(x)S(x) + xz_{TE}. \tag{1}$$

This function is composed of the class function $C(x)$ and the shape function $S(x)$ plus a term $xz_{TE}$ which characterizes the trailing-edge thickness. The $x$ is defined in the interval [0,1] and the $x, z$ coordinates of the airfoil are normalized. Two functions are needed to represent the upper and lower surfaces of an airfoil, respectively. The class function, as shown in Eq. (2), defines the basic shape of the geometry.

$$C(x) = x^{N_1}(1-x)^{N_2}, \tag{2}$$

where $N_1$ and $N_2$ are hyper parameters, different combinations of $N_1$ and $N_2$ in the class function define a variety of general classes of geometries. For airfoils which usually have a blunt nose and sharp trailing edge, $N_1$ equals to 0.5 and $N_2$ equals to 1 [19]. The shape function delineates the geometric characteristics based on the class function and facilitates the systematic generation of a diverse set of component airfoil geometries, which can be scaled to represent a wide variety of airfoil geometries. Typically, Bernstein polynomials are adopted in the shape function, the mathematical formulation of the shape function using Bernstein polynomials is presented below:

$$S(x) = \sum_{i=0}^{N} A_i S_{i,N}(x), \tag{3}$$

$$S_{i,N}(x) = \frac{N!}{i!(N-i)!} x^i (1-x)^{N-i}, \tag{4}$$

where $A_i$ are parameters determined by regression (such as least squares regression) from discrete $x, z$ coordinates. Since two functions are needed to represent the upper and lower surfaces of an airfoil, the number of parameters for a $N^{\text{th}}$ order CST parameterization method to represent an airfoil is $2(N+1)$. The curves of $S(x/c)$ and $S(x/c)C(x/c)$ ($N_1 = 0.5$, $N_2 = 1$) for $2^{\text{nd}}$ and $8^{\text{th}}$ CST parameterization method are shown in Fig. 1, the $x/c$ and $z/c$ here refer to the normalized $(x, z)$ coordinates by chord length $c$.

Taking the derivative of the $i^{\text{th}}$ component in the $N^{\text{th}}$ order $S(x)C(x)$ and setting it to zero, the position of the maximum value occurs at:

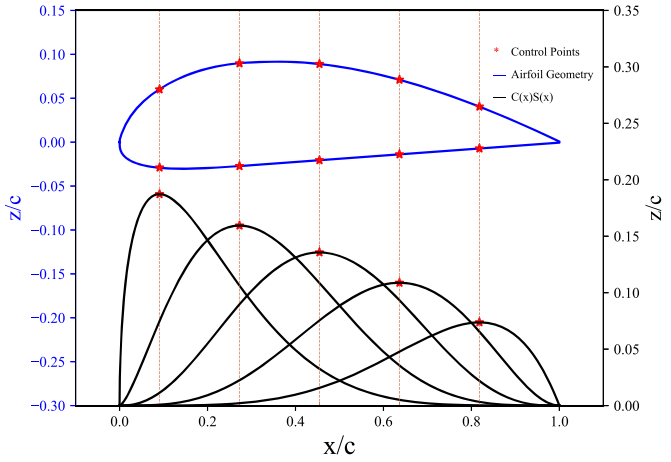



**Fig. 2.** Distribution of control points for a 4th order CST parameterization method on the surface of the Clark Y airfoil.

$$x_{C(x)S_{i,N}(x)_{max}} = \frac{N_1 + i}{N_1 + N_2 + n}. \tag{5}$$

This result demonstrates that the peaks of these polynomial curves are uniformly distributed within the interval [0, 1]. Since each peak exerts the most significant influence on the overall geometry at its location [18], the control points of the CST parameterization method are uniformly distributed, as illustrated in Fig. 2.

Although the CST parameterization method is considered one of the most effective parameterization methods for airfoil design optimization, researches [2,3] have concluded that at least 25 parameters (corresponding to more than 11th order for the CST parameterization method) are necessary to accurately represent no less than 80% of airfoils in large airfoil libraries.Furthermore, in critical regions, such as the leading edge of the airfoil, the low-order CST parameterization method often struggles to accurately represent the geometry within the tolerances required in wind tunnel tests. While the high-order CST parameterization method suffers ill-conditioning parameterization matrix [22,25,26]. However, modifying the CST parameterization method to overcome these problems necessitates a high level of expertise in both mathematics and aerodynamics, posing challenges for researchers with insufficient domain knowledge.

## 3. Framework of the LLM-based symbolic regression method for enhancing CST parameterization method

To address the above issues in the CST parameterization method, this paper proposes an LLM-based symbolic regression method to enhance the performance of the CST parameterization method by replacing $x$ with a feature transformation function $t(x)$ in the class function and shape function as shown below:

$$z_{\text{new}} = C(t(x)) \cdot S(t(x)) + x \cdot z_{\text{TE}}, \tag{6}$$

where $x$ and $z_{\text{new}}$ represent the $x$ and $z$ coordinates of the points on an airfoil, respectively. $z_{\text{TE}}$ represents the trailing-edge thickness, which can refer to either the upper or lower surface of the airfoil, depending on which surface is parameterized. The proposed method leverages the advanced natural language processing and reasoning capabilities of LLM, alongside the evolutionary framework's proficiency in effectively searching complex design space to identify global optimal solutions. Consequently, it facilitates the modification of airfoil parameterization methods with less domain knowledge and provides flexibility in the selection of basic mathematical operators. The framework of the proposed LLM-based symbolic regression method is illustrated in Fig. 3 and the process is demonstrated below:

1. Initialization: The pre-trained LLM is utilized to generate an initial generation containing M functions based on the **Initialization Prompt** provided in natural language.

2. Evaluation: The generated functions are applied to modify the CST parameterization method. The performance of the modified CST parameterization methods is evaluated by the weighted sum of fitting errors in the geometry recovery test of airfoils in the training set.

3. Evolution: Four **Evolution Prompts**, each offering distinct trade-offs between exploration and exploitation, are used to generate new functions on the basis of previous generation. In each generation, each strategy is executed M times to produce M new functions, resulting in a total of 4M new functions derived from the previous generation. Detailed descriptions of the Evolution Prompts are provided in the Appendix A.

4. Population management: The top M functions (ranked by fitness value) from the current population are selected to form the population for the subsequent generation.

5. Iteration: Steps 1 to 4 are repeated iteratively until satisfactory functions are designed.

Compared with GP-based symbolic regression method, the proposed LLM-based symbolic regression method does not require any specification of basic mathematical operators (such as +, -, ×, ÷, sin, cos, log, etc.) before design. Thus, it provides a larger search space containing more basic mathematical operators. Additionally, the strong semantic understanding and language processing capabilities of LLM allow designers to give mathematical insights (such as the need for functions to be continuous and differentiable) in natural language, eliminating the need to represent these preferences using complex codes.

During design, each airfoil in the training set was fitted by the modified CST parameterization method using least squares regression. The order of the Bernstein polynomial was set to 6, 8 and 10 respectively, which are the orders commonly utilized by the CST parameterization method for airfoil design optimization. The objective value returned for sorting and selecting individuals is the weighted sum of the fitting error for each airfoil in the training set. The formulation for calculating the fitting error of a single airfoil is presented below:

$$\text{fitting error} = \sum_{k}^{K} \sum_{i=1}^{n} w|z_i - z_{\text{order}=k}(x_i)|, \quad K = 6, 8, 10, \tag{7}$$

where $n$ is the total number of points used to define the upper and lower surfaces of an airfoil ($n = n_{\text{upper}} + n_{\text{lower}}$), $z_i$ is the real value for the coordinate $z$ and $z(x_i)$ is calculated by the modified CST parameterization method, $K$ is the order of the Bernstein polynomial, which is also the order of the modified CST parameterization method. The variable $w$ represents the weight of the fitting error associated with $x_i$. The value of $w$ is calculated by Eq. 8 [3,44] since higher precision is required at the leading edge of the airfoil.

$$w = \begin{cases} 2 & \text{if } x_i < 0.2 \\ 1 & \text{if } x_i \geq 0.2. \end{cases} \tag{8}$$

## 4. Experiments for enhancing the CST parameterization method through LLM-based symbolic regression method

This section provides detailed introduction to the experiments for designing the feature transformation function aimed at enhancing the CST parameterization method. The pre-trained LLM GPT-4o-mini was utilized for generating and evolving functions. During evolution, the maximum number of generations was set to 20, the population size was set to 15 and four evolution prompts were used simultaneously. The entire framework was executed on a single CPU (AMD Ryzen 7 5800H with Radeon Graphics). Three experiments were conducted with varying levels of mathematical insights in prompts. Due to the randomness in the generation of the initial population, each experiment with different settings was independently repeated five times. Besides, in order to further demonstrate the advantages of using LLM for symbolic regression,

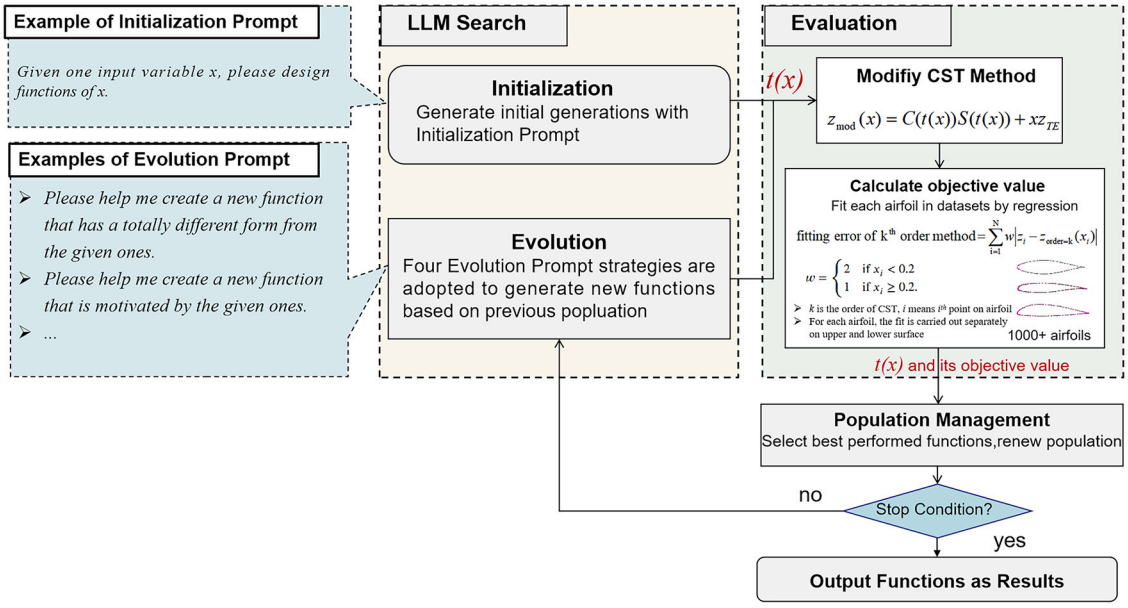



**Fig. 3.** The framework of the proposed LLM-based symbolic regression method for enhancing the performance of the CST parameterization method.

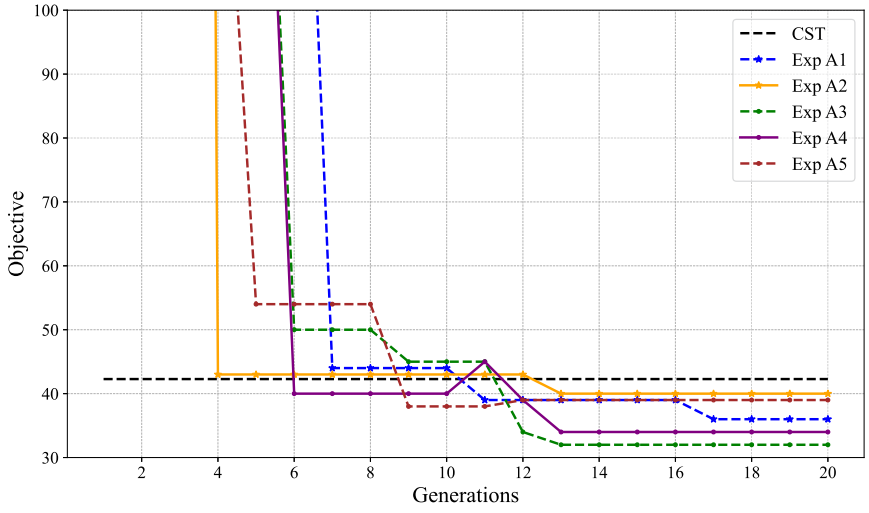



**Fig. 4.** The convergence histories of five independent experiments in Experiment A.

we also conducted GP-based symbolic regression to design the feature transformation functions. Comparison of the performance between two symbolic regression methods is also discussed in this section.

### 4.1. Datasets

The training set was constructed by randomly selecting 1000 airfoils from the University of Illinois Urbana-Champaign (UIUC) airfoil library (https://m-selig.ae.illinois.edu/ads/coorddatabase.html). This library contains NACA, Eppler, RAE, Clark and many other airfoil families, making it widely used for testing the performance of the parameterization methods [2,3]. The testing set consisted 500 airfoils randomly selected from the remaining airfoils in the UIUC airfoil library after excluding the training set. The airfoils in the datasets were stored using $x, z$ coordinates in the .dat file and discretized between 24 and 205 points.

### 4.2. Experiment A: simple prompt without mathematical insights

Experiment A aims to evaluate whether our framework can design satisfactory feature transformation functions using simple prompts without any expert knowledge. The Initialization Prompt was intentionally kept straightforward, without any mathematical insights as follows: ***Given one input variable*** $x$***, please design functions of*** $x$**.** The convergence histories of five independent experiments in the Experiment A are shown in Fig. 4. The objective value of the baseline CST parameterization method (which is 42.28) is also plotted in Fig. 4 for comparison. Table 1 summarizes the performance of functions designed in five independent experiments in Experiment A. The top five functions designed in Experiment A and their objective values are summarized in Table 2. Results demonstrate that all five independent experiments converged, but some failed to achieve satisfactory results despite outperforming the baseline CST parameterization method. This highlights the low design efficiency of the Experiment A.

Based on the results presented above, we found that basic mathematical operators, such as max and sigmoid, which are typically not considered in traditional symbolic regression tasks, were utilized in the functions designed. This highlights the flexibility in selecting basic operators of the proposed LLM-based symbolic regression method, in contrast to GP-based symbolic regression, which requires the predefinition of basic mathematical operators. However, due to the sim-

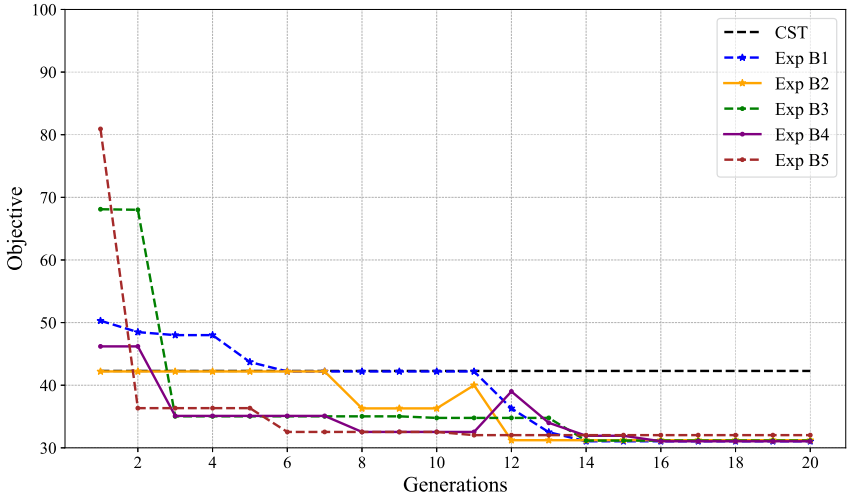



**Fig. 5.** The convergence histories of five independent experiments in Experiment B.

**Table 1**
The performance of functions in the 20th generation of five independent experiments in Experiment A.

| No. experiment | Number of functions outperform baseline CST method | Number of functions with NaN fitness value |
|---|---|---|
| 1 | 2 | 6 |
| 2 | 1 | 10 |
| 3 | 2 | 8 |
| 4 | 3 | 9 |
| 5 | 1 | 7 |

plicity of the Initialization Prompt, the proposed LLM-based symbolic regression method generated a lot of invalid functions, including $y = -x$ and $y = \text{random}(x)$ (as shown in Table 3), which resulted in an objective value of Not a Number (NaN). This issue led to low design efficiency and needs be addressed. In addition, duplicated functions with different expressions for $x$ in the interval $[0, 1]$ were generated even in the same population, some examples are presented in Table 4.

### 4.3. Experiment B: add simple mathematical insights to prompt

In Experiment B, simple insights were incorporated into the Initialization prompt to eliminate invalid functions generated during evolution. These insights were obtained by letting LLM summarize the well-performing functions designed in the Experiment A. All other settings, including objective function, population size, remained the same with those in Experiment A. The Initialization prompt used in Experiment B is: ***Given one input variable $x$, please design functions of $x$. The function must be continuous, differentiable, monotonically increasing and positive for x in the interval [0,1]. When x is 1.0 or 0.0, the corresponding value calculated by the designed function should also be 1.0 or 0.0.***

The convergence histories of five independent experiments conducted in Experiment B are presented in Fig. 5. Compared with Experiment A, all five experiments designed better functions with faster convergence speed after adding insights summarized by LLM from Experiment A. This indicates that by incorporating simple mathematical insights through natural language, the design efficiency of the LLM-based symbolic regression method can be significantly improved. Table 5 summarizes the performance of functions in the 20th generation of Experiment B. Table 6 demonstrates the performance of top five functions designed.

It can be concluded that by incorporating simple mathematical insights into the initialization prompt, the number of invalid functions significantly reduced in five independent experiments, making the proposed LLM-based symbolic regression method able to design more feasible functions with enhanced performance. This indicates that the LLM effectively comprehended mathematical insights conveyed through natural language, thereby facilitating the expression of designers' requirements more intuitively than GP-based symbolic regression methods, which often involve a complex coding process.

### 4.4. Experiment C: add well-performed functions to the initial population

In this experiment, the top 15 functions designed in Experiment B were added to the initial population. All other settings remained the same with those in Experiments B including prompts, number of generations and population size. The convergence histories of five independent experiments conducted in Experiment C are presented in Fig. 6. These results indicate that by adding well-performed functions to the initial population, the LLM-based symbolic regression method successfully designed better functions compared with Experiment B. Table 7 summarizes the performance of functions designed in the last generation of Experiment C. All functions in the last generation of five independent experiments can effectively enhance the performance of the CST parameterization method and the NaN objective values disappeared. The performance of top five functions designed in Experiment C are presented in Table 8. Compared with Experiment B, these functions exhibited better performance, indicating that a well-chosen initial population from previous design results can effectively enhance the efficiency of the proposed LLM-based symbolic regression method.

Notably, we found that LLM do have issues such as hallucinations during evolution, which increases the likelihood of generating unsuitable functions (see those returned NaN fitness value in Table 1 and Table 5). However, our proposed LLM-based symbolic regression method combines LLM with evolutionary framework. The fitting error of the modified CST parameterization method on the training set was used as the fitness value to sort and select functions. In this process, even if LLM produces an unsuitable function, it will be filtered out due to its poor performance, thus not affecting the final results. Taking Experiment A and B as examples, invalid functions are observed even in the last generation as shown in Table 5. However, these invalid functions were eliminated or not adopted as design results due to their poor performance.

### 4.5. Comparison with GP-based symbolic regression method

In this section, to give detailed comparison between Genetic Programming (GP)-based symbolic regression method and the proposed

**Table 2**
Top five functions designed in Experiment A using simple prompts.

| Functions designed (in Python code form) | Objective value |
|---|---|
| $ta1(x) = np.sqrt(x)$ | 32.54 |
| $ta2(x) = np.cbrt(x ** 2)$ | 34.06 |
| $ta3(x) = np.loglp(np.abs(x))/np.log(2)$ | 36.34 |
| $ta4(x) = np.abs(np.sin(x))/np.max(np.abs(np.sin(x)))$ | 39.97 |
| $ta5(x) = (1/sigmoid(x)) - np.min((1/sigmoid(x))))/(np.max((1/sigmoid(x))) - np.min((1/sigmoid(x)))$ | 40.76 |

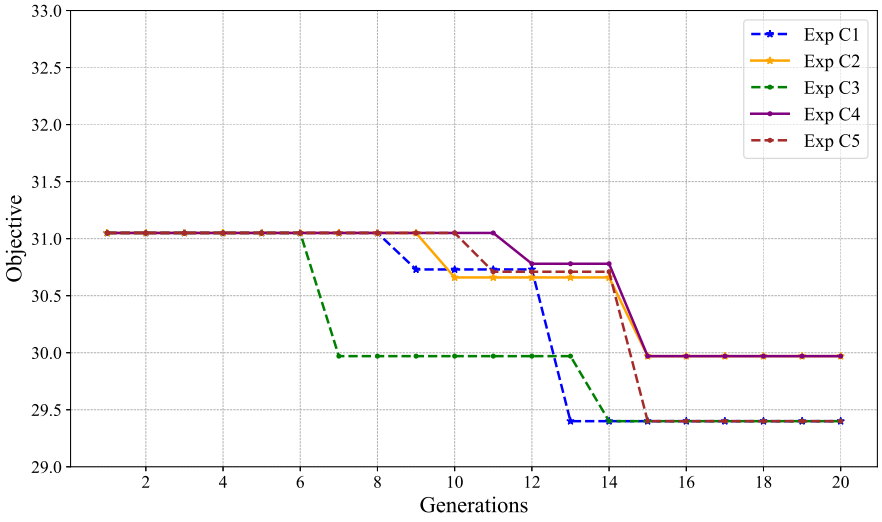



**Fig. 6.** The convergence histories of five independent experiments in Experiment C.

**Table 3**
Some invalid functions designed in Experiment A.

| Problem | Functions designed (in Python code form) |
|---|---|
| negative value | $t(x) = -x$<br>$t(x) = x ** 3 - 3 * x ** 2 + 3 * x - 1$<br>$t(x) = -np.exp(np.abs(x))$ |
| not continuous | $t(x) = random(x)$<br>$t(x) = np.sin(x) + np.random.rand(* x.shape) * -10$ |

**Table 4**
Examples of duplicated functions with different expressions in Experiment A.

| Simplified function for $x$ in [0,1] | Functions designed (in Python code form) |
|---|---|
| $y = x^{0.5}$ | $t(x) = np.sqrt(x)$<br>$t(x) = np.sqrt(np.abs(x)) * np.sign(x)$<br>$t(x) = np.abs(np.sqrt(x))$<br>$t(x) = np.power(x, 0.5)$ |
| $y = \frac{ln(x+1)}{ln2}$ | $t(x) = np.log(np.abs(x) + 1)/np.log(2)$<br>$t(x) = np.loglp(x)/np.loglp(1)$<br>$t(x) = np.loglp(abs(x))/np.log(2)$ |

**Table 5**
The performance of functions in the 20th generation of five independent experiments in Experiment B.

| No. experiment | Number of functions outperform baseline CST method | Number of functions with NaN fitness value |
|---|---|---|
| 1 | 12 | 0 |
| 2 | 9 | 1 |
| 3 | 7 | 1 |
| 4 | 13 | 0 |
| 5 | 12 | 0 |

**Table 6**
Top five functions designed in Experiment B.

| Functions designed (in python code form) | Objective value |
|---|---|
| $tb1(x) = np.sqrt(x) * (1 + (x - 1)/2)$ | 31.05 |
| $tb2(x) = np.sqrt(x) + 0.1 * (x ** 2 - x)$ | 31.16 |
| $tb3(x) = (x ** (1/3)) * (1 + (x - 1)/3)$ | 31.22 |
| $tb4(x) = (x ** (1/3)) * (1 + (x ** (1/3) - 1)/2)$ | 31.93 |
| $tb5(x) = np.sqrt(x) * (1 + (np.sqrt(x) - 1)/2)$ | 32.03 |

**Table 7**
The performance of functions in the 20th generation of five independent experiments in Experiment C.

| No. experiment | Number of functions outperform baseline CST method | Number of functions with NaN fitness value |
|---|---|---|
| 1 | 15 | 0 |
| 2 | 15 | 0 |
| 3 | 15 | 0 |
| 4 | 15 | 0 |
| 5 | 15 | 0 |

**Table 8**
Top five functions designed in Experiment C.

| Functions designed (in Python code form) | Objective value |
|---|---|
| $tc1(x) = 0.5 * x + 0.5 * np.sign(x) * np.abs(x) ** (1/3)$ | 29.40 |
| $tc2(x) = (x ** (1/3))/(1 + (1 - (x ** (1/3))))$ | 29.97 |
| $tc3(x) = (x ** (1.0/2.0)) * (1.0 + (x ** (1.0/2.0) - 1.0)/3.0)$ | 30.66 |
| $tc4(x) = np.sqrt(x) * (1.0 + (np.sqrt(x) - 1.0)/4.0)$ | 30.71 |
| $tc5(x) = np.cbrt(x) * (1.0 + 2.0 * (np.cbrt(x) - 1.0)/3.0)$ | 30.73 |

LLM-based symbolic regression method, we repeated the aforementioned experiments using GP-based symbolic regression method [39,46]. The GP-based symbolic regression is an evolutionary algorithm for discovering mathematical expressions. Similarly to the LLM-based sym-

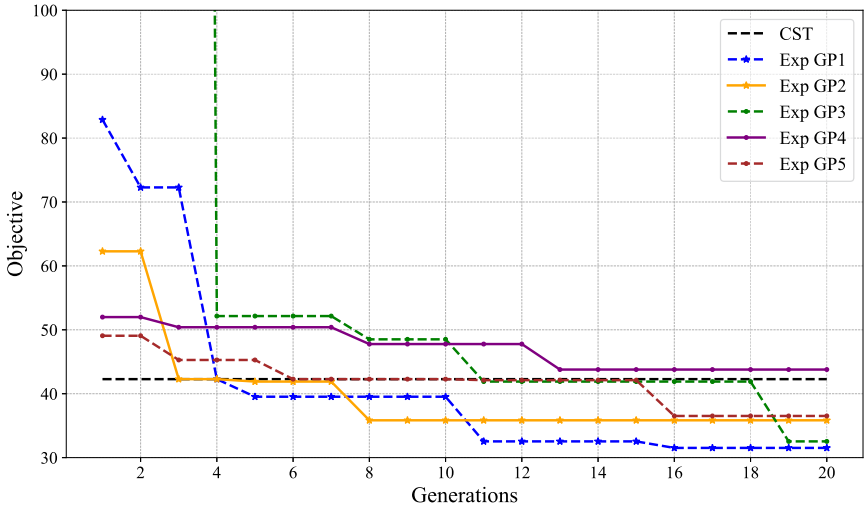



**Fig. 7.** The convergence histories of five independent experiments using GP-based symbolic regression method.

**Table 9**
The performance of functions in the 20th generation of five independent experiments using GP-based symbolic regression method.

| No. experiment | Number of functions outperform baseline CST method | Number of functions with NaN fitness value |
|---|---|---|
| 1 | 4 | 6 |
| 2 | 2 | 9 |
| 3 | 5 | 8 |
| 4 | 2 | 7 |
| 5 | 1 | 7 |

**Table 10**
Top five functions designed by GP-based symbolic regression method.

| Functions designed (in Python code form) | Objective value |
|---|---|
| $tgp1(x) = x ** (1.0/2.0) * (1 + (x ** (1.0/2.0) - 1)/2)$ | 32.23 |
| $tgp2(x) = x ** (1.0/2.0)$ | 32.54 |
| $tgp3(x) = 0.433 * (1.0 - (1.0 - x) ** 3.0) + 0.567 * x$ | 35.59 |
| $tgp4(x) = 0.252 * (1.0 - (1.0 - x) ** 3.0) + 0.748 * x$ | 35.82 |
| $tgp5(x) = x ** (1.0/4.0) * (1.0 + (x ** (1.0/4.0) - 1.0)/4.0)$ | 43.54 |

bolic regression method, GP-based symbolic regression method also leverages evolutionary framework to optimize both the structure and parameters of functions. This method begins with the random generation of an initial population of functions, often encoded as expression trees. Genetic operators such as selection, crossover and mutation are applied to evolve these expressions over generations based on their fitness.

To ensure the fairness of comparison, the population size and the number of generations were set to 15 and 20, respectively. The probabilities for crossover and mutation were adoptively adjusted based on the population's fitness [47,48] and were set to 0.8 and 0.1 at the beginning of evolution. Additionally, to avoid generating overly complex functions and causing overfitting, we limited the depth of the expression tree to 4. The basic mathematical operators include add, sub, mu, div, sin, cos, tan, exp, log, sqrt, pow, abs, max, min, neg, sigmoid and tanh were used to generate functions. These operators were chosen because they were adopted in the results generated by LLM-based symbolic regression method. Although some of them (such as max, min, neg and sigmoid) are not commonly used in typical GP-based symbolic regression, they were still included in our experiments to ensure fairness of comparison. Five independent experiments were conducted and the convergence histories are demonstrated in Fig. 7. The design results are demonstrated in Table 9 and Table 10.

**Table 11**
Comparison of experiments conducted using GP-based and LLM-based symbolic regression methods.

| Experiment | Mathematical insight | Fitness of top 3 functions |
|---|---|---|
| Experiment GP (GP-based) | No (Need to code) | 32.23, 32.54, 35.59 |
| Experiment A (LLM-based) | No | 32.54, 34.06, 36.34 |
| Experiment B (LLM-based) | Yes (In natural language) | 31.05, 31.16, 31.22 |
| Experiment C (LLM-based) | Yes (In natural language) | 29.40, 29.97, 30.66 |

These results demonstrate that the GP-based symbolic regression method achieved results comparable to those in Experiment A. However, when more mathematical insights are added to the prompt, the GP-based symbolic regression method was not able to design functions that outperform results in Experiment B and C as shown in Table 11. In the last generation, similar to those found in Experiment A, we observed the presence of invalid and redundant equations as shown in Table 9.

We posit that this is because the mathematical insights in prompts enable the LLM to consciously filter the generated functions, thereby enhancing the design efficiency. In contrast to the straightforward natural language expressions utilized in the LLM-based symbolic regression method to convey mathematical insights, the GP-based symbolic regression method necessitates meticulous constraints or additional processing (such as taking the absolute value of the function) to articulate these requirements, rendering it less convenient. Furthermore, the GP-based symbolic regression algorithm lacks the ability to automatically summarize features of well-performing results, which is a capability inherent in the LLM-based method.

Based on above analysis, the advantages of the LLM-based symbolic regression method over the GP-based symbolic regression method are as follows: (1). The LLM-based symbolic regression method does not require predefined mathematical operators. (2). The LLM has strong natural language processing and reasoning capabilities, enabling designers to easily convey their requirements without the need to resort to coding. (3). The LLM can automatically summarize insights from design results and inspire designers, a capability that the GP-based symbolic regression method lacks.

### 4.6. *Mathematical features of designed functions*

Analyzing the mathematical features of the top five functions designed in Experiment C, the curves of these functions are illustrated in Fig. 8. The designed functions are concave functions with larger curvature near the leading edge of the airfoil (where $x = 0$). The $C(x)S(x)$ (taking the 4th order as example) before and after transformation using

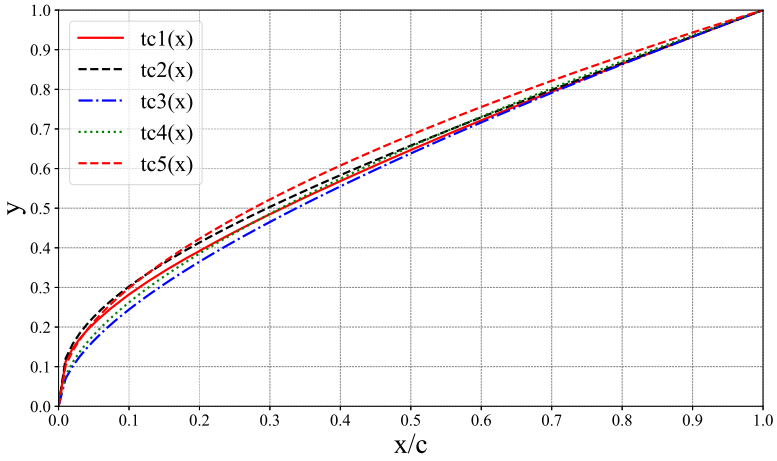
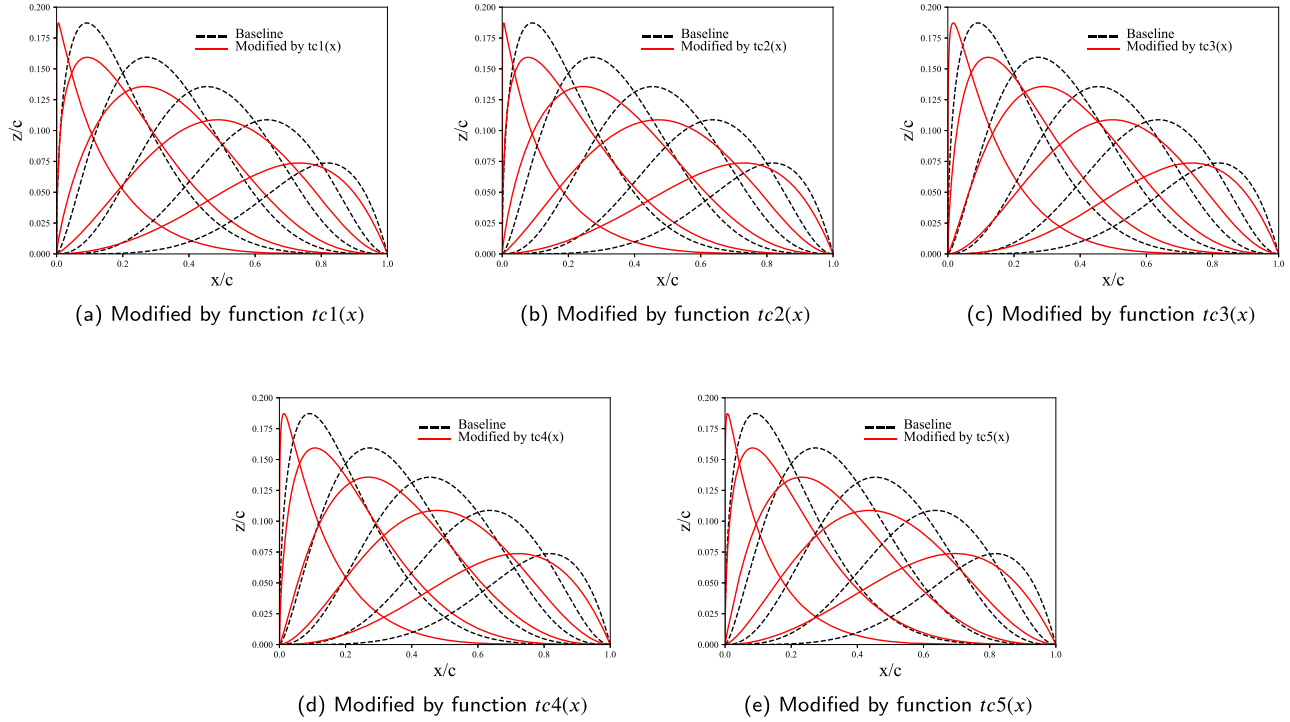



**Fig. 8.** The curves of the top five functions designed in Experiment C.




(a) Modified by function $tc1(x)$ (b) Modified by function $tc2(x)$ (c) Modified by function $tc3(x)$

(d) Modified by function $tc4(x)$ (e) Modified by function $tc5(x)$

**Fig. 9.** Curves of C(x)S(x) (order = 4) before and after transformation using top five functions designed.

the designed functions are demonstrated in Fig. 9. It is observed that the peaks of all the polynomial terms have shifted forward after transformation. This indicates that the modified CST parameterization method exhibits enhanced control capabilities near the leading edge of the airfoil as more control points are concentrated in that region (see Fig. 10). This discovery aligns with the conclusions drawn in previous work on modifying the CST parameterization method by experts [24,49].

Note that by moving the position of control points towards the leading edge of the airfoil, the fitting accuracy at the trailing edge of the airfoil could be affected. However, since the leading edge is pivotal in determining aerodynamic performance and is more sensitive to the fitting accuracy, enhancing the fitting accuracy at the leading edge while only slightly compromising the accuracy at the trailing edge is typically considered acceptable for most airfoils. This conclusion will be further validated in Section 5.

From all experiments conducted in this section, we have demonstrated advantages of employing LLM for symbolic regression, including no need to predefine basic mathematical operators, the avoidance of complex programming and the capability to let designers articulate their requirements more intuitively using natural language. In summary, the LLM-based symbolic regression method successfully designed feature transformation functions with clear and interpretable mathematical characteristics. In addition to enhancing the performance of the CST parameterization method, it also offers valuable mathematical insights for further enhancing the effectiveness of the airfoil parameterization method.

## 5. Validation of the CST parameterization method modified by LLM (CST-LLM)

### 5.1. Approximation accuracy of airfoil geometries

In this section, the function $tc1(x) = 0.5 \cdot x + 0.5 \cdot \text{np.sign}(x) \cdot |x|^{1/3}$, which is the best function designed in Experiment C, was used to modify the CST parameterization method. The modified CST parameterization

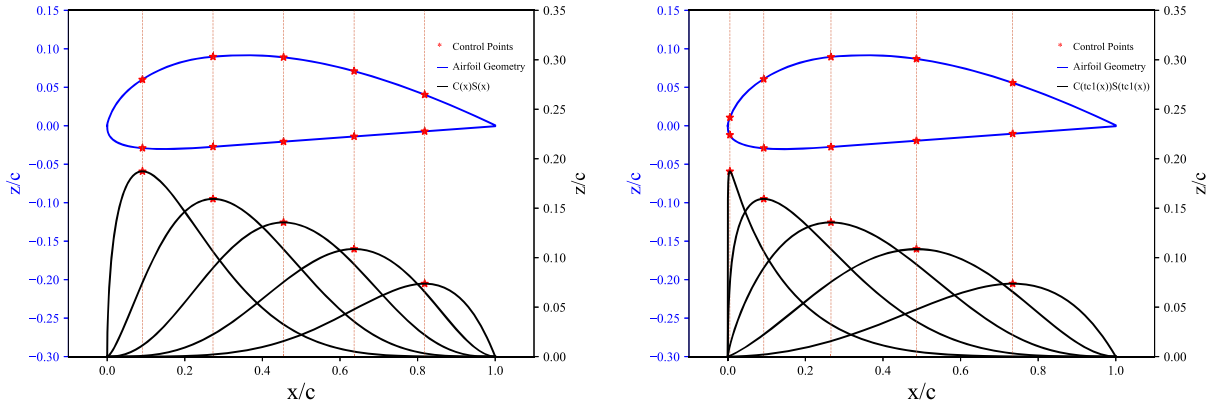



**Fig. 10.** Control points distribution (order = 4) of the CST parameterization method before and after transformation using function $tc1(x)$.

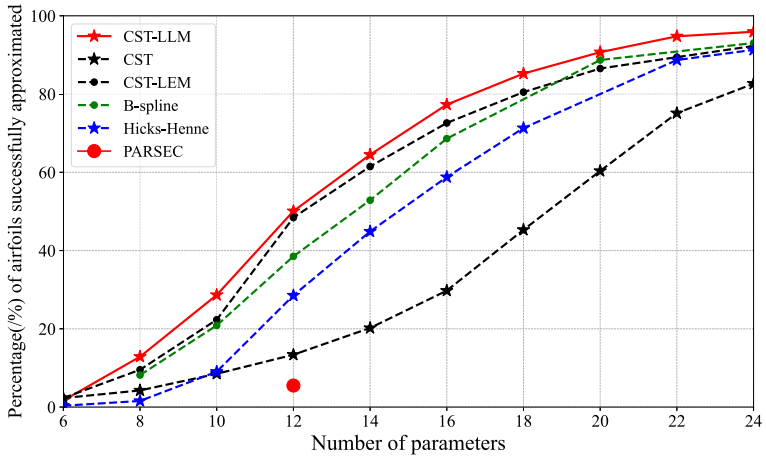



**Fig. 11.** Approximating accuracy comparison of the CST-LLM parameterization method in the testing set with other state-of-the-art parameterization methods.

method by function $tc1(x)$ is referred to as the CST-LLM parameterization method in the following discussion. The approximation accuracy of the CST-LLM parameterization method on the testing set was evaluated for orders ranging from $4^{th}$ to $12^{th}$ and compared with the CST parameterization method [18] as well as other state-of-the-art parameterization methods including the B-spline [16,17], PARSEC [14,15], Hicks-Henne [10] and CST-LEM method. The approximation accuracy is defined as the proportion of airfoils in the testing set that can be accurately fitted within the wind tunnel test tolerance [18,22], as shown in Eq. (9). This metric reflects whether a parameterization method can more effectively represent the geometries of different kinds of airfoils.

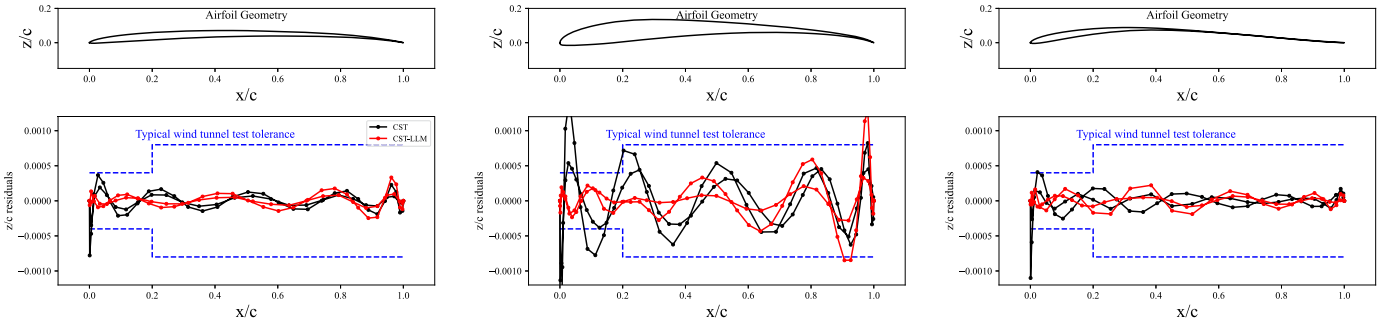
$$Tolerance = \begin{cases} 3.5 \times 10^{-4} & \text{if } x_i < 0.2 \\ 7 \times 10^{-4} & \text{if } x_i \geq 0.2 \end{cases} \tag{9}$$

The relationship between the number of parameters and the order of the CST parameterization method is: Number of parameters = $2(order + 1)$. The settings for other parameterization methods were based on the conclusions from reference [2,3] as shown in the Appendix B. This is to ensure that each parameterization method achieved its optimal performance for testing. For constructive methods like B-spline, PARSEC and CST-LEM, the least squares regression was adopted to calculate the parameters from airfoil coordinates. For deformative methods such as Hicks-Henne, the NACA 0012 airfoil was used as the initial airfoil geometry and parameters were used to deform the NACA 0012 airfoil to fit different airfoils. The parameters in the deformative methods were also calculated by least squares regression. The approximating accuracies in the testing set of the CST-LLM, CST and other parameterization methods are shown in Fig. 11. These results demonstrate that the CST-LLM parameterization method, which uses the feature transformation function $tc1(x)$ designed in Experiment C, achieved significant improvements in approximation accuracy compared with the CST parameterization method.

### 5.2. *Case studies of fitting accuracy for various types of airfoils*

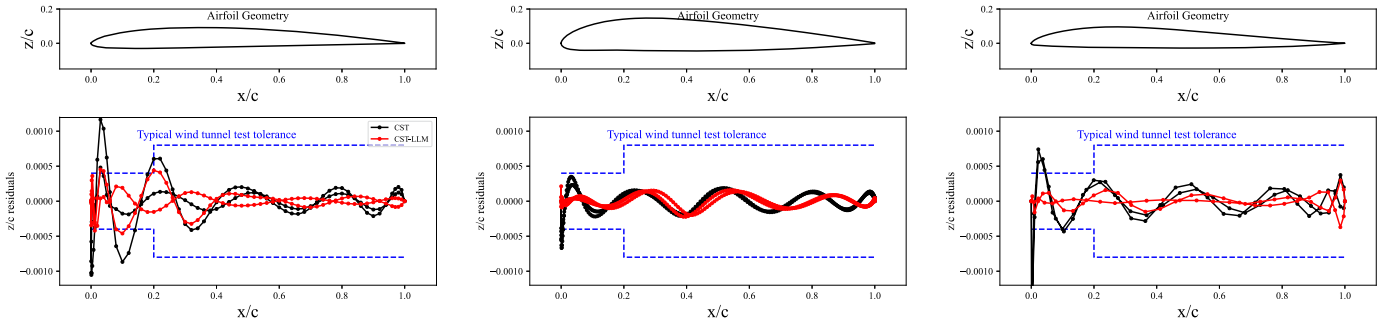
For a more comprehensive investigation, this section presents the enhancements in fitting accuracy of the CST-LLM parameterization method on four categories of airfoils with different geometric features. These airfoils are utilized in a variety of scenarios, including the wings of supersonic and transonic aircraft, vertical tails and propellers. The first category is the cambered thin airfoil, this kind of airfoil features large camber and relatively small thickness. The second category is the flat-bottom airfoil, this kind of airfoil is characterized by a flat lower surface and large camber. The third category is the symmetry airfoil, which has symmetric upper and lower surfaces. The fourth category is the airfoil with aft-loading at the trailing edge of the lower surface. The order of the CST-LLM and CST parameterization methods was set to 8. The results are presented in Fig. 12 to Fig. 15 and demonstrate that the fitting accuracies of the CST-LLM parameterization method exhibited significant improvement compared with the CST parameterization method, particularly at the leading edge of airfoils.

However, for airfoils like S 1223 and EPPLER 476, a slight decrease in the fitting accuracy was observed at the trailing edge. In Fig. 16, we



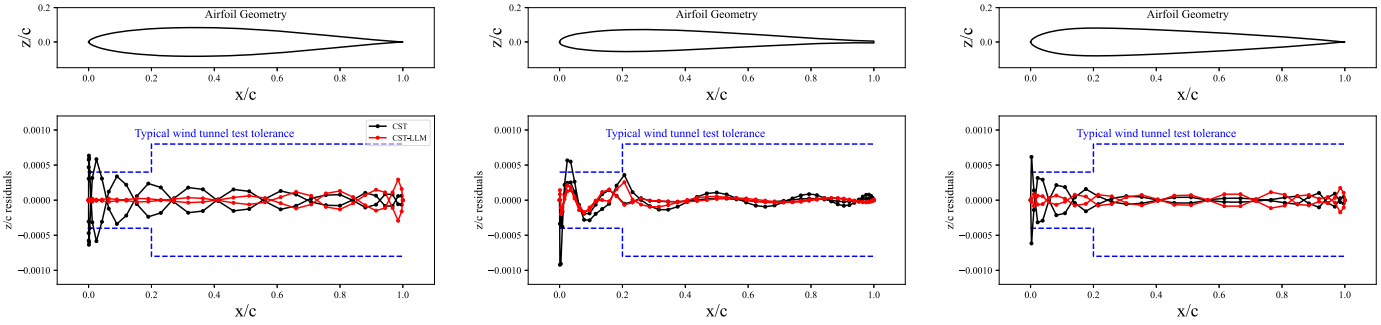

(a) E 63 (b) S 1223 (c) EPPLER 378

**Fig. 12.** Fitting accuracy comparison of cambered airfoils.



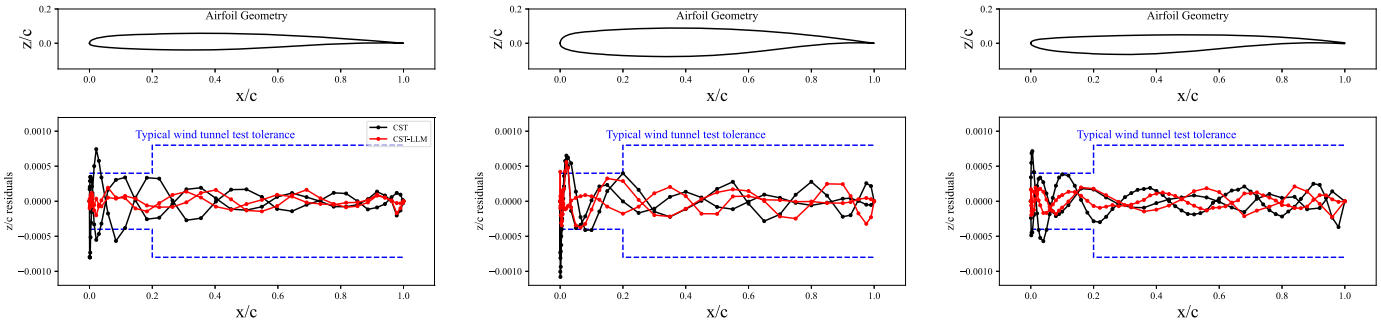

(a) Clark Y (b) NACA 65-019 (c) EPPLER 637

**Fig. 13.** Fitting accuracy comparison of flat-bottom airfoils.



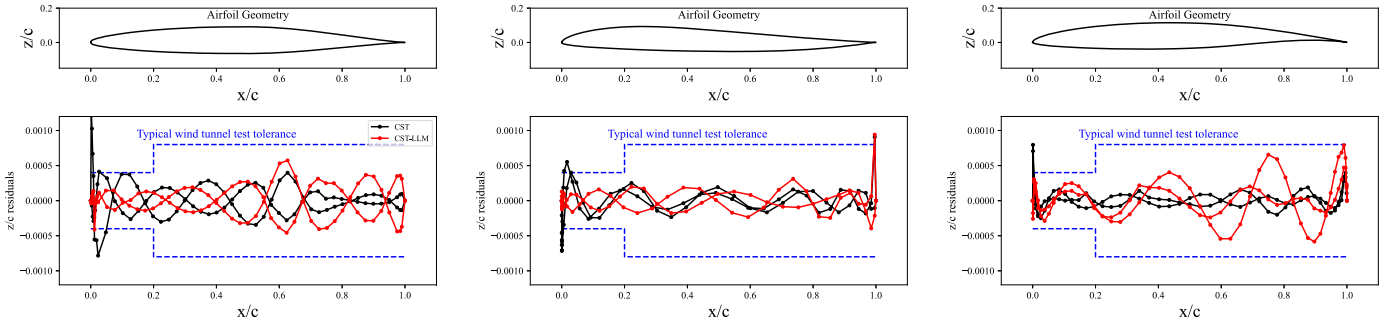

(a) EPPLER 476 (b) OAF 128 (c) E474

**Fig. 14.** Fitting accuracy comparison of symmetry airfoils.




(a) RAE 5214 (b) NLR 7301 (c) NPL 9510

**Fig. 15.** Fitting accuracy comparison of airfoils with aft-loading.




(a) NACA66-215 (b) EPPLER 340 (c) FX62-K-153

**Fig. 16.** Fitting accuracy comparison of airfoils where CST-LLM is less accurate at the trailing edge.

**Table 12**
Design space for design optimization of the RAE 5214 airfoil.

| Order | Method | Boundary | Parameters |
|---|---|---|---|
| 4 | CST | Upper | 0.32, 0.08, 0.37, 0.17, 0.25, -0.07, -0.01, -0.12, 0.06, 0.02 |
| | | Lower | 0.11, 0.03, 0.12, 0.06, 0.08, -0.2, -0.04, -0.35, 0.02, 0.01 |
| 4 | CST-LLM | Upper | -0.001, 0.32, 0.081, 0.46, 0.30, -0.01, -0.04, -0.05, -0.11, 0.17 |
| | | Lower | -0.003, 0.11, 0.027, 0.15, 0.10, -0.02, -0.12, -0.16, -0.34, 0.06 |
| 6 | CST | Upper | 0.33, 0.07, 0.41, 0.03, 0.41, 0.15, 0.25, -0.07, -0.01, -0.11, -0.01, -0.08, 0.11, 0.01 |
| | | Lower | 0.11, 0.02, 0.14, 0.01, 0.14, 0.05, 0.08, -0.21, -0.03, -0.32, -0.02, -0.25, 0.04, 0.0 |
| 6 | CST-LLM | Upper | 0.0, 0.24, 0.17, 0.2, 0.35, 0.34, 0.33, 0.0, -0.04, -0.05, -0.01, -0.16, 0.06, 0.05 |
| | | Lower | -0.01, 0.08, 0.06, 0.07, 0.12, 0.11, 0.11, -0.01, -0.11, -0.15, -0.04, -0.49, 0.02, 0.02 |

present more cases where the CST-LLM parameterization method performed badly. It is evident that all airfoils exhibited significant geometric variations at the trailing edge. This phenomenon can be attributed to the forward displacement of the peaks in the Bernstein polynomial in the CST-LLM parameterization method, as depicted in Fig. 10, which results in a reduction of the fitting accuracy at the airfoil's trailing edge. However, this decrease in accuracy was observed in only a small number of airfoils. For the majority of airfoils, the leading edge serves as the primary acceleration region of the flow, necessitating a higher fitting accuracy. Therefore, the CST-LLM parameterization method developed in this paper exhibits superior performance compared with the CST parameterization method in most cases and is capable of accurately representing diverse types of airfoils with distinctly different geometric features.

### 5.3. Applications in aerodynamic design optimization of airfoil

For further verification, design optimizations were performed using both CST and CST-LLM parameterization methods. Two different orders (4 and 6) were utilized in the parameterization method. Note that for the $4^{\text{th}}$ order CST (or CST-LLM) parameterization method, there are 10 design variables for design optimization, while the $6^{\text{th}}$ order CST (or CST-LLM) parameterization method involves 14 design variables. To ensure the reliability of the results, the experiments were conducted 5 times independently under each setting.

The Mach number was set to 0.735 and the Reynolds number was set to $6.5\times10^6$. The design optimization aimed to minimize the drag coefficient $C_d$ without reducing the lift coefficient $C_l$, thickness and the pitching moment coefficient $C_m$ of the airfoil. The aerodynamic characteristics of the airfoil were calculated using a CFD solver based on the Reynolds-averaged Navier-Stokes equations (RANS) [50], with the Spalart-ALLMaras model [51] chosen as the turbulence model. The objective and constraints are shown below.

$$\begin{aligned} &\min \quad \text{obj} = C_d \\ &\text{w.r.t.} \quad A_i \in [A_{i,\min}, A_{i,\max}] \\ &\text{s.t.} \quad C_l \geq C_{l,\text{base}} \\ &\qquad C_m \geq C_{m,\text{base}} \\ &\qquad t \geq t_{\text{base}}. \end{aligned} \tag{10}$$

where $C_{l,\text{base}}$ and $C_{m,\text{base}}$ refer to the $C_l$ and $C_m$ of the baseline airfoil with $C_{l,\text{base}} = 0.88$, $C_{m,\text{base}} = -0.078$. $t_{\text{base}}$ here refers to the thickness of the baseline airfoil and equals to 0.095. The surrogate optimization method using kriging model was applied for airfoil design optimization [52–54]. Initial samples were generated using the Latin Hypercube Sampling (LHS) [55] method. The EI [56] and MSP [57] infill-sampling criteria were used to update the surrogate model. The maximum iteration was set to 160 for each design optimization. The upper and lower boundaries of the design space were obtained by applying a 50% perturbation to the CST (or CST-LLM) parameters fitted from the RAE 5214 airfoil, which are illustrated in Fig. 17 and Table 12. Note that for the CST parameterization method, regardless of whether a feature transformation function is used, the corresponding shapes remain identical if the parameters are scaled proportionally. This ensures that the design space obtained by proportionally scaling the parameters from both CST and CST-LLM parameterization methods remained the same in the design optimization.

The convergence histories of design optimizations employing the CST and the CST-LLM parameterization methods with different orders are shown in Fig. 18. The results are summarized in Table 13. It is evident that the results designed using the CST-LLM parameterization consistently outperformed the CST parameterization method, demonstrating the effectiveness of the CST-LLM parameterization method in airfoil design optimization. Furthermore, in some experiments, the design optimization using $4^{\text{th}}$ order CST-LLM parameterization method achieved results comparable to the $6^{\text{th}}$ order CST parameterization method, further validating the superior performance of the CST-LLM parameterization method in airfoil design optimization.

**Table 13**
The results of design optimization using the CST and CST-LLM parameterization method in 5 independent experiments.

| Airfoil | Parameterization method | Order | Cd/cts in 5 independent experiments |
|---|---|---|---|
| RAE 5214 | base geometry | - | 364 |
| Design using 4th order CST | CST | 4 | 123(best),124,123,124,125 |
| Design using 4th order CST-LLM | CST-LLM | 4 | 113(best),113,114,113,113 |
| Design using 6th order CST | CST | 6 | 114(best),120,114,117,121 |
| Design using 6th order CST-LLM | CST-LLM | 6 | 108(best),112,116,108,110 |

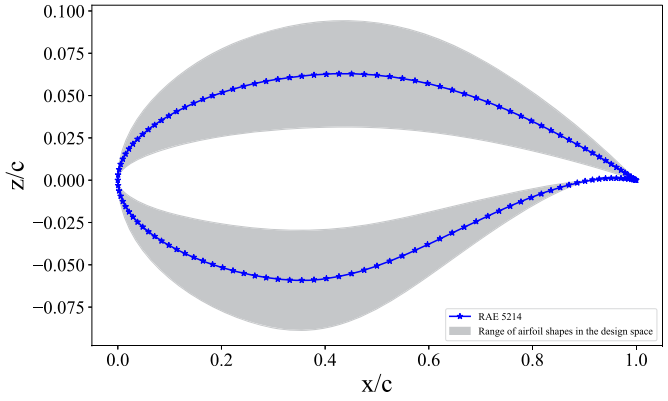



**Fig. 17.** Geometry of the RAE 5214 airfoil and the range of airfoil geometries in the design space.

Analyzing the best results obtained using $6^{th}$ and $4^{th}$ order CST and CST-LLM parameterization methods, the geometries of the designed airfoils are demonstrated in Fig. 19. The pressure coefficient distributions and the pressure contours are illustrated in Fig. 20 and Fig. 21, respectively. The pressure coefficient distributions indicate that the airfoil designed using the CST-LLM parameterization method can more effectively reduce the shockwave intensity compared with airfoil designed using the CST parameterization method especially when order = 4.

## 6. Conclusions and future work

This paper proposes an LLM-based symbolic regression method for enhancing the performance of the CST parameterization method. The natural language is used as prompt for guiding LLM to generate and evolve functions. The evolutionary framework is used to optimize the generated functions. The CST-LLM parameterization method is proposed by modifying CST parameterization method using feature transformation functions designed by the LLM-based symbolic regression method. Verification using geometry recovery test and application in design optimization are conducted. Conclusions drawn from this paper are as follows.

(1) Through experiments incorporating varying levels of mathematical insights in prompts, the proposed symbolic regression method is proven to be able to leverage the reasoning and synthesis capabilities of LLM. This allows designers to articulate their requirements and preferences in natural language, thus eliminating the complex coding processes typically needed for traditional symbolic regression methods. When facing completely unknown problems, designers can start with the most basic instructions to guide the LLM for generating functions. When well-performed results are obtained, designers can collaborate with LLM to summarize insights from these results (like the process in Experiment A to Experiment B). This not only aids in the discovery of new scientific theorems but also lowers the entry barriers for researchers in specific research fields.

(2) Compared with GP-based symbolic regression method, the proposed LLM-based symbolic regression method does not require the specification of basic mathematical operators at the beginning of the design. Instead, it relies on the LLM to automatically generate and evolve functions, offering a higher degree of freedom and a larger search space compared with traditional symbolic regression methods. Once the feature transformation function is designed, it can be directly applied in aerodynamic design optimization by simply transforming the x-coordinates of the airfoil before using CST parameterization method with nearly no additional computational cost.

(3) The CST-LLM parameterization method, generated using feature transformation function $tc1(x)$ designed by the LLM-based symbolic regression method, improves the fitting accuracy of the airfoil geometries especially at the leading edge by shifting the peak of the terms in $C(x)S(x)$ towards $x = 0$. This is in accordance with the stringent fitting accuracy requirements at the leading edge of airfoil.

(4) In most cases of geometry recovery test, the CST-LLM parameterization method outperforms other state-of-the-art parameterization

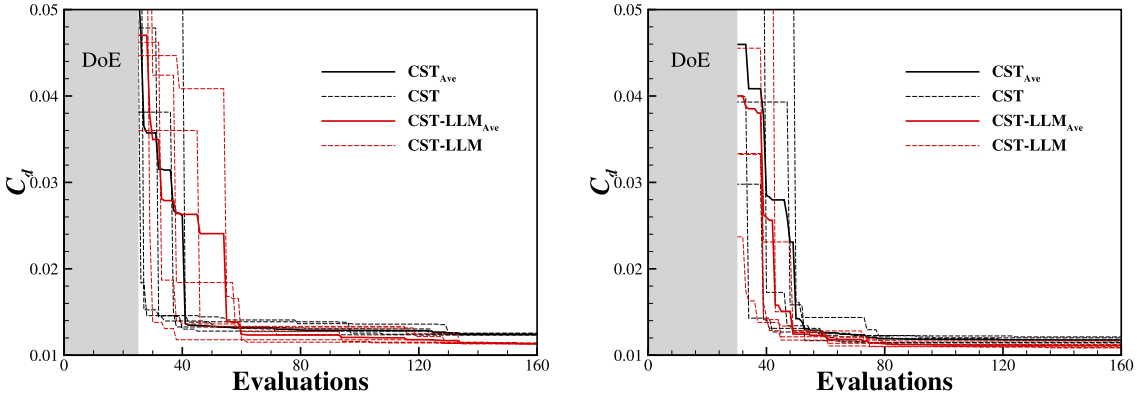
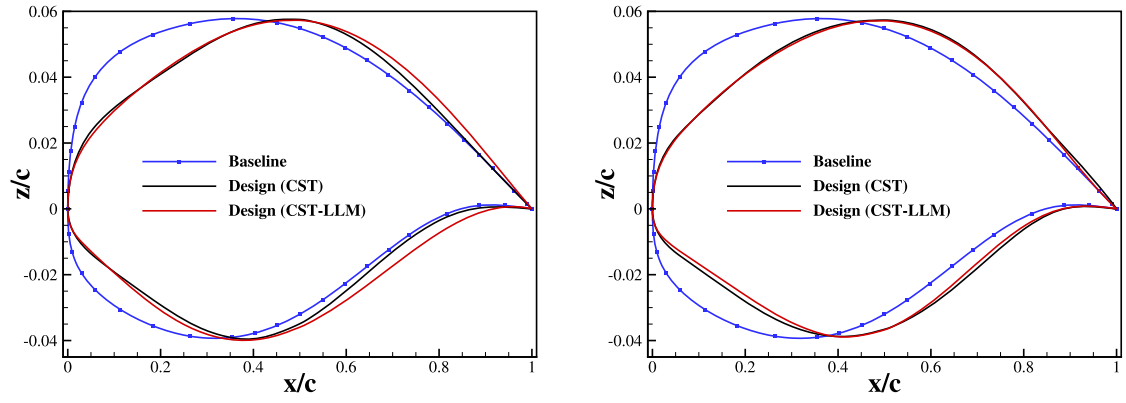
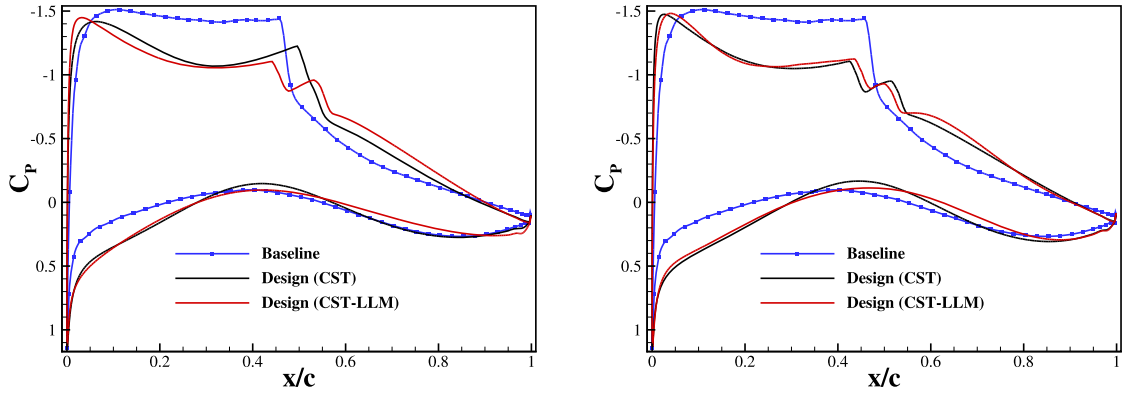



**Fig. 18.** The convergence histories of design optimizations using CST and CST-LLM parameterization methods (each run 5 times independently).



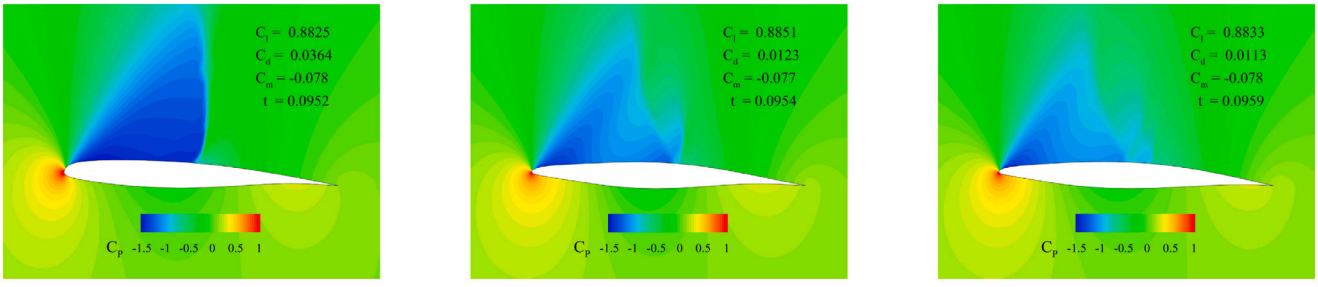

(a) order=4 (b) order=6

**Fig. 19.** Airfoils designed using CST and CST-LLM parameterization methods in the design optimization.

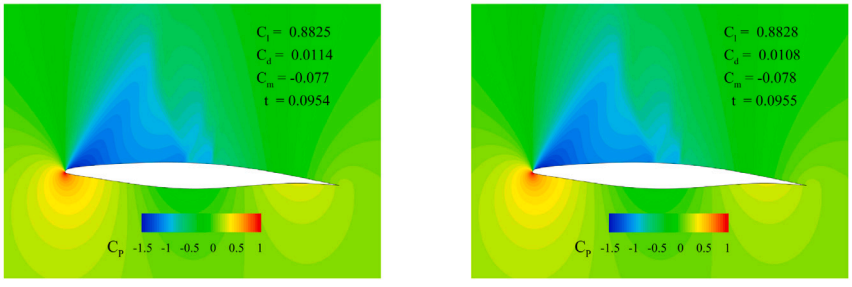



(a) order=4 (b) order=6

**Fig. 20.** The pressure coefficient distribution of airfoils designed using CST and CST-LLM parameterization methods in the design optimization.




(a) Baseline (RAE 5214) (b) Design using CST, order=4 (c) Design using CST-LLM, order=4




(d) Design using CST, order=6 (e) Design result using CST-LLM, order=6

**Fig. 21.** The pressure contour of airfoils designed using CST and CST-LLM parameterization methods in the design optimization.

methods in accurately representing different kinds of airfoil geometries, offering a more complete design space for airfoil design optimization. In the aerodynamic design optimization of RAE 5214 airfoil, the CST-LLM parameterization method consistently outperformed the baseline CST parameterization method and even achieved better results using fewer control parameters. These results further demonstrate the effectiveness of the CST-LLM parameterization method as well as the proposed LLM-based symbolic regression method.

However, the LLM-based symbolic regression framework proposed in this paper still has some shortcomings. From the design results in Table 2, 6 and 8, while the LLM successfully designed well-performed feature transformation functions using an evolutionary framework, the coefficients in these functions remained overly simplistic (with most coefficients being integer). This highlights the limitation of LLM in achieving more precise design results [40]. Therefore, it is essential to combine the proposed LLM-based symbolic regression method with other optimization algorithms (i.e., the Broyden-Fletcher-Goldfarb-Shanno algorithm [58]) to facilitate a bilevel search for coefficients. This will be a key focus of our future research. Furthermore, we will investigate the applications of LLM for reflective analysis throughout symbolic regression to facilitate automatic assessment of features of the designed functions, thereby enhancing the automation of the design process.

Additionally, based on the observations from Fig. 16, while the CST-LLM parameterization method developed in this paper has shown a significant improvement in representing the leading edge geometry of airfoils, there is a small probability of accuracy degradation at the trailing edge of airfoils with substantial geometric variations. This issue can be addressed by modifying the structure of the training set (including more airfoils with significant variations of geometries at the trailing edge) and designing new feature transformation functions. In the future, we will also focus on creating unique feature transformation functions customized for specific types of airfoils. In addition, as the ability of LLM becomes stronger and stronger, we will also consider using our proposed LLM-based symbolic regression approach to design new parameterization methods instead of modifying existing parameterization method.

## CRediT authorship contribution statement

**Kefeng Zheng:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Yiheng Wang:** Writing – review & editing, Validation, Software, Methodology. **Fei Liu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization. **Qingfu Zhang:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Wenping Song:** Writing – review & editing, Validation, Software, Resources, Methodology, Investigation, Conceptualization.

## Funding

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. CityU11217325].

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. Evolution prompts

**Evolution Prompt (EP1)**

Given one input variable x, please design functions of x.
I have 2 pairs of functions as follows.
<Function 1>
<Function 2>
Please help me create a new function that has a totally different form from the given ones.
(For Experiment B and C, summarized by LLM from results in Experiment A) The function must be continuous, differentiable, monotonically increasing and positive for x in the interval [0,1]. When x is 1.0 or 0.0, the corresponding value calculated by the designed function should also be 1.0 or 0.0.

**Evolution Prompt (EP2)**

Given one input variable x, please design functions of x.
I have 2 pairs of functions as follows.
<Function 1>
<Function 2>
Please help me create a new function that is motivated by the given ones.
(For Experiment B and C, summarized by LLM from results in Experiment A) The function must be continuous, differentiable, monotonically increasing and positive for x in the interval [0,1]. when x is 1.0 or 0.0, the corresponding value calculated by the designed function should also be 1.0 or 0.0.

**Evolution Prompt (EP3)**

Given one input variable x, please design functions of x.
I have one function as follows.
<Function 1>
Please help me create a new function that is a revision of the given one.
(For Experiment B and C, summarized by LLM from results in Experiment A) The function must be continuous, differentiable, monotonically increasing and positive for x in the interval [0,1]. when x is 1.0 or 0.0, the corresponding value calculated by the designed function should also be 1.0 or 0.0.

**Evolution Prompt (EP4)**

Given one input variable x, please design functions of x.
I have one function as follows.
<Function 1>
Please help me create a new function that has different parameter settings of the given one.
(For Experiment B and C, summarized by LLM from results in Experiment A) The function must be continuous, differentiable, monotonically increasing and positive for x in the interval [0,1]. when x is 1.0 or 0.0, the corresponding value calculated by the designed function should also be 1.0 or 0.0.

## Appendix B. Introduction to other parameterization methods compared in Section 5

### B.1. CST-LEM parameterization method

The CST-LEM (Leading-Edge Modification) parameterization method is a modified version of the CST parameterization method. This method aims to enhance the approximation of airfoil geometries with large leading-edge camber by adding an additional function in the shape function as shown in Eq. (11). This will add an additional parameter $A_{N+1}$ and to ensure consistency in the number of parameters during testing, we reduced the order of Bernstein polynomial in CST-LEM parameterization method by 1 when compare CST-LLM with CST-LEM parameterization method.

$$S(x) = \sum_{i=0}^{N} A_i S_{i,N}(x) + A_{N+1} x^{0.5}(1-x)^{N-0.5}. \tag{11}$$

### B.2. B-spline parameterization method

The B-spline parameterization method is a widely used method for producing piecewise polynomial curves. This method relies on a set of basis functions, the coefficients are defined spatially by a set of discrete control points $P_i \in \mathbb{R}^3$. The curve, parametrized by the scalar $u \in [u_0, u_m]$, is given by Eq. (12):

$$C(u) = \sum_{i=0}^{n-1} N_{i,p}(u) P_i, \tag{12}$$

where the $n = m - p$ basis functions of order $p$ are given by Eq. (13):

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1, & u_i \le u < u_{i+1} \\ 0, & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{13}$$

In the test, each airfoil was represented by two B-splines. For each B-spline, $\mathbf{P}_0$ was fixed at the leading edge $(0,0)$, $\mathbf{P}_{n+1}$ was at the trailing edge, $\mathbf{P}_1$ was vertically aligned with the leading edge and other control points $\mathbf{P}_i$ were distributed on a half-cosine scale between $(0,1)$ chordwise and only allowed to vary vertically. The control points were defined in Eq. (14).

$$\mathbf{P}_0 = (0,0), \quad \mathbf{P}_i = \left( \frac{1}{2}\left[1 - \cos\left(\frac{\pi(i-1)}{n+1}\right)\right], a_i \right), \quad \mathbf{P}_{n+1} = (1, z_{te}), \tag{14}$$

where $a_i$ represents the vertical design variables and $z_{te}$ denotes the vertical coordinate of the trailing edge position. Research [2] have concluded that 3rd, 4th and 5th order B-splines exhibited comparable accuracy when utilizing more than 25 design variables. However, for cases involving fewer design variables, 5th-order B-splines shown marginally superior performance. Consequently, this paper employed 5th order B-spline curves in the test.

### B.3. PARSEC parameterization method

The PARSEC parameterization method defines the airfoil geometry analytically based on geometry properties such as the upper/lower crest position, max thickness, leading-edge radius and boat-tail angle. This is done by proposing that the upper and lower surfaces should be defined by the 6th order polynomials as shown in Eq. (15):

$$z(x) = \sum_{i=1}^{6} a_i x^{i-0.5}. \tag{15}$$

**Table 14**
The parameters used for the PARSEC method and their definitions.

| Parameter | Definition |
|---|---|
| Upper leading-edge radius ($r_{leu}$) | $r_{leu} = a_1$ |
| Lower leading-edge radius ($r_{lel}$) | $r_{lel} = b_1$ |
| Upper crest point ($Z_{up}$) | $Z_{up} = z_{up}(X_{up})$ |
| Lower crest point ($Z_{lo}$) | $Z_{lo} = z_{lo}(X_{lo})$ |
| Position of upper crest ($X_{up}$) | $z'_{up}(X_{up}) = 0$ |
| Position of lower crest ($X_{lo}$) | $z'_{lo}(X_{lo}) = 0$ |
| Upper crest curvature ($Z_{XXup}$) | $z''_{up}(X_{up}) = Z_{XXup}$ |
| Lower crest curvature ($Z_{XXlo}$) | $z''_{lo}(X_{lo}) = Z_{XXlo}$ |
| Trailing edge offset ($Z_{TE}$) | $z_{lo}(1) = Z_{TE}$ |
| trailing-edge thickness ($\Delta Z_{TE}$) | $z_{up}(1) = Z_{TE} + \Delta Z_{TE}$ |
| Trailing edge angle ($\alpha_{TE}$) | $z'_{up}(1) = -\tan\left(\alpha_{TE} + \frac{\beta_{TE}}{2}\right)$ |
| Boat-tail angle ($\beta_{TE}$) | $z'_{lo}(1) = -\tan\left(\alpha_{TE} - \frac{\beta_{TE}}{2}\right)$ |

Twelve equations, subject to twelve free parameters were used to define the characteristics of an airfoil, their definitions are given in Table 14.

### B.4. Hicks-Henne parameterization method

The Hicks-Henne parameterization method uses a base shape $z_{\text{base}}$ plus a linear combination of a set of $N$ basis functions defined between 0 and 1 to determine the geometry. Each surface is defined by Eq. (16) and usually the basis functions [10] is the sine functions as shown in Eq. (17).

$$z(x) = z_{\text{base}}(x) + \sum_{i=0}^{N} A_i \phi_i(x), \tag{16}$$

$$\phi_i(x) = \sin(t_i \pi x^{m_i}), \quad m_i = \ln(0.5)/\ln(x_{M_i}), \tag{17}$$

where $x_{M_i}$ is the location of the maxima of the basis function and $t_i$ controls the width of the functions. Each bump function is therefore defined by three variables. Various combinations of fixing and optimizing these variables have been performed. In our test, conclusions from existing research [2] were adopted where the $A_i$ was determined by least squares regression and $M_i$ was calculated from Eq. (18) with $t_i = 1$.

$$x_{M_i} = \frac{1}{2}\left[1 - \cos\left(\frac{i\pi}{n+1}\right)\right], \quad i = 1, \ldots, n. \tag{18}$$

## Data availability

Data will be made available on request.

## References


[1] N.P. Salunke, R.A.J. Ahamad, S. Channiwala, Airfoil parameterization techniques: a review, Am. J. Mech. Eng. 2 (4) (2014) 99–102, https://doi.org/10.12691/ajme-2-4-1.

[2] D.A. Masters, N.J. Taylor, T. Rendall, C.B. Allen, D.J. Poole, Review of aerofoil parameterisation methods for aerodynamic shape optimisation, in: 53rd AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 2015.

[3] D.A. Masters, N.J. Taylor, T.C.S. Rendall, C.B. Allen, D.J. Poole, Geometric comparison of aerofoil shape parameterization methods, AIAA J. 55 (5) (2017) 1575–1589, https://doi.org/10.2514/1.j054943.

[4] X. Lu, J. Huang, L. Song, J. Li, An improved geometric parameter airfoil parameterization method, Aerosp. Sci. Technol. 78 (2018) 241–247, https://doi.org/10.1016/j.ast.2018.04.025.

[5] P. Castonguay, S. Nadarajah, Effect of shape parameterization on aerodynamic shape optimization, in: 45th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2007.

[6] S. Nadarajah, P. Castonguay, A. Mousavi, Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization, in: 18th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2007.

[7] V. Sripawadkul, M. Padulo, M. Guenov, A comparison of airfoil shape parameterization techniques for early design optimization, in: 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, American Institute of Aeronautics and Astronautics, 2010.

[8] J. Vassberg, A. Jameson, Influence of Shape Parameterization on Aerodynamic Shape Optimization, von Karman Institute for Fluid Dynamics, 2018.
[9] J. Hoyos, J.H. Jímenez, C. Echavarría, J.P. Alvarado, Airfoil shape optimization: comparative study of meta-heuristic algorithms, airfoil parameterization methods and Reynolds number impact, IOP Conf. Ser., Mater. Sci. Eng. 1154 (1) (2021) 012016, https://doi.org/10.1088/1757-899x/1154/1/012016.
[10] R. Hicks, P. Henne, Wing design by numerical optimization, in: Aircraft Systems and Technology Meeting, American Institute of Aeronautics and Astronautics, 1977.
[11] T.C.S. Rendall, C.B. Allen, Unified fluid–structure interpolation and mesh motion using radial basis functions, Int. J. Numer. Methods Eng. 74 (10) (2007) 1519–1559, https://doi.org/10.1002/nme.2219.
[12] J. Samareh, Aerodynamic shape optimization based on free-form deformation, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, 2004.
[13] A. Sóbester, A.I. Forrester, Aircraft Aerodynamic Design: Geometry and Optimization, John Wiley Sons, 2014.
[14] H. Sobieczky, Parametric Airfoils and Wings, 1999, pp. 71–87.
[15] R. Derksen, T. Rogalsky, Bezier-PARSEC: an optimized aerofoil parameterization for design, Adv. Eng. Softw. 41 (7–8) (2010) 923–930, https://doi.org/10.1016/j.advengsoft.2010.05.002.
[16] V. Braibant, C. Fleury, Shape optimal design using B-splines, Comput. Methods Appl. Mech. Eng. 44 (3) (1984) 247–267, https://doi.org/10.1016/0045-7825(84)90132-4.
[17] H. Sobieczky, Geometry Generator for CFD and Applied Aerodynamics, Springer, Vienna, 1997, pp. 137–157.
[18] B. Kulfan, J. Bussoletti, "Fundamental" parameteric geometry representations for aircraft component shapes, in: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2006, p. 6948.
[19] B.M. Kulfan, Recent extensions and applications of the 'CST' universal parametric geometry representation method, Aeronaut. J. 114 (1153) (2010) 157–176, https://doi.org/10.1017/s0001924000003614.
[20] H.-Y. Wu, S. Yang, F. Liu, H.-M. Tsai, Comparisons of three geometric representations of airfoils for aerodynamic optimization, in: 16th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2003.
[21] Z. Xiaoping, D. Jifeng, L. Weiji, Z. Yong, Robust airfoil optimization with multi-objective estimation of distribution algorithm, Chin. J. Aeronaut. 21 (4) (2008) 289–295, https://doi.org/10.1016/s1000-9361(08)60038-2.
[22] X. Guan, Z. Li, B. Song, A study on CST aerodynamic shape parameterization method, Acta Aeronaut. Astronaut. Sin. 33 (4) (2012) 625–633, doi:CNKI:11-1929/V.20111011.1411.005.
[23] K.M. Selvan, On the effect of shape parameterization on aerofoil shape optimization, Int. J. Res. Eng. Technol. 4 (2) (2015) 123–133, https://doi.org/10.15623/ijret.2015.0402016.
[24] B. Kulfan, Modification of CST airfoil representation methodology, http://www.brendakulfan.com/docs/CST8.pdf, 2009.
[25] M. Ceze, M. Hayashi, E. Volpe, A study of the CST parameterization characteristics, in: 27th AIAA Applied Aerodynamics Conference, American Institute of Aeronautics and Astronautics, 2009.
[26] H.M. Hajdik, A. Yildirim, E. Wu, B.J. Brelje, S. Seraj, M. Mangano, J.L. Anibal, E. Jonsson, E.J. Adler, C.A. Mader, G.K.W. Kenway, J.R.R.A. Martins, pyGeo: a geometry package for multidisciplinary design optimization, J. Open Source Softw. 8 (87) (2023) 5319, https://doi.org/10.21105/joss.05319.
[27] H. Naveed, A.U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A comprehensive overview of large language models, arXiv preprint, arXiv:2307.06435, 2023, https://doi.org/10.48550/arXiv.2307.06435.
[28] F. Liu, Y. Yao, P. Guo, Z. Yang, X. Lin, X. Tong, M. Yuan, Z. Lu, Z. Wang, Q. Zhang, A systematic survey on large language models for algorithm design, arXiv preprint, arXiv:2410.14716, 2024, https://doi.org/10.48550/arXiv.2410.14716.
[29] F. Liu, T. Xialiang, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, Q. Zhang, Evolution of heuristics: towards efficient automatic algorithm design using large language model, in: Forty-First International Conference on Machine Learning, 2024.
[30] F. Liu, X. Tong, M. Yuan, Q. Zhang, Algorithm evolution using large language model, arXiv preprint, arXiv:2311.15249, 2023, https://doi.org/10.48550/arXiv.2311.15249.
[31] S. Chithrananda, G. Grand, B. Ramsundar, ChemBERTa: large-scale self-supervised pretraining for molecular property prediction, arXiv preprint, arXiv:2010.09885, 2020, https://doi.org/10.48550/arXiv:2010.09885.
[32] D.A. Boiko, R. MacKnight, B. Kline, G. Gomes, Autonomous chemical research with large language models, Nature 624 (7992) (2023) 570–578, https://doi.org/10.1038/s41586-023-06792-0.
[33] S. Li, S. Moayedpour, R. Li, M. Bailey, S. Riahi, L. Kogler-Anele, M. Miladi, J. Miner, D. Zheng, J. Wang, A. Balsubramani, K. Tran, M. Zacharia, M. Wu, X. Gu, R. Clinton, C. Asquith, J. Skaleski, L. Boeglin, S. Chivukula, A. Dias, F.U. Montoya, V. Agarwal, Z. Bar-Joseph, S. Jager, CodonBERT: large language models for mRNA design and optimization, https://doi.org/10.1101/2023.09.09.556981, Sep. 2023.
[34] B. Ni, M.J. Buehler, MechAgents: large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge, Extrem. Mech. Lett. 67 (2024) 102131, https://doi.org/10.1016/j.eml.2024.102131.
[35] Y. Zhang, K. Zheng, F. Liu, Q. Zhang, Z. Wang, AutoTurb: using large language models for automatic algebraic turbulence model discovery, Phys. Fluids 37 (1) (Jan. 2025), https://doi.org/10.1063/5.0247759.
[36] Microsoft Research AI4Science, Microsoft Azure Quantum, The impact of large language models on scientific discovery: a preliminary study using GPT-4, arXiv preprint, arXiv:2311.07361, 2023, https://doi.org/10.48550/arXiv.2311.07361.
[37] P. Ma, T.-H. Wang, M. Guo, Z. Sun, J.B. Tenenbaum, D. Rus, C. Gan, W. Matusik, LLM and simulation as bilevel optimizers: a new paradigm to advance physical scientific discovery, arXiv preprint, arXiv:2405.09783, 2024, https://doi.org/10.48550/arXiv:2405.09783.
[38] P. Guo, Q. Zhang, X. Lin, CoEvo: continual evolution of symbolic solutions using large language models, arXiv preprint, arXiv:2412.18890, 2024, https://doi.org/10.48550/arXiv:2412.18890.
[39] N. Makke, S. Chawla, Interpretable scientific discovery with symbolic regression: a review, Artif. Intell. Rev. 57 (1) (Jan. 2024), https://doi.org/10.1007/s10462-023-10622-0.
[40] M. Merler, K. Haitsiukevich, N. Dainese, P. Marttinen, In-context symbolic regression: leveraging large language models for function discovery, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), Association for Computational Linguistics, 2024, pp. 589–606.
[41] M. Du, Y. Chen, Z. Wang, L. Nie, D. Zhang, Large language models for automatic equation discovery of nonlinear dynamics, Phys. Fluids 36 (9) (Sep. 2024), https://doi.org/10.1063/5.0224297.
[42] B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M.P. Kumar, E. Dupont, F.J.R. Ruiz, J.S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, A. Fawzi, Mathematical discoveries from program search with large language models, Nature 625 (7995) (2023) 468–475, https://doi.org/10.1038/s41586-023-06924-6.
[43] R. Zhang, F. Liu, X. Lin, Z. Wang, Z. Lu, Q. Zhang, Understanding the Importance of Evolutionary Search in Automated Heuristic Design with Large Language Models, Springer Nature, Switzerland, 2024, pp. 185–202.
[44] B.M. Kulfan, Universal parametric geometry representation method, J. Aircr. 45 (1) (2008) 142–158, https://doi.org/10.2514/1.29958.
[45] H. Su, L. Gu, C. Gong, Research on geometry modeling method based on three-dimensional CST parameterization technology, in: 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, 2015.
[46] J. Koza, Genetic programming as a means for programming computers by natural selection, Stat. Comput. 4 (2) (Jun. 1994), https://doi.org/10.1007/bf00175355.
[47] A.E. Eiben, Z. Michalewicz, M. Schoenauer, J.E. Smith, Parameter Control in Evolutionary Algorithms, Springer Berlin Heidelberg, 2007, pp. 19–46.
[48] S. Silva, E. Costa, Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories, Genet. Program. Evol. Mach. 10 (2) (2009) 141–179, https://doi.org/10.1007/s10710-008-9075-9.
[49] D.D. Marshall, Creating exact Bezier representations of CST shapes, in: 21st AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2013.
[50] S.B. Pope, Turbulent flows, Meas. Sci. Technol. 12 (11) (2001) 2020–2021.
[51] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 1992.
[52] Z.H. Han, SurroOpt: a generic surrogate-based optimization code for aerodynamic and multidisciplinary design, in: Proceedings of ICAS, vol. 2016, 2016, 2016-0281.
[53] C.Z. Xu, Z.H. Han, K.S. Zhang, H. Zeng, C. Gong, Z. Zhou, A novel multi-fidelity cokriging model assisted by multiple non-hierarchical low-fidelity datasets, in: Structural and Multidisciplinary Optimization, Springer Science and Business Media LLC, 2024.
[54] C.Z. Xu, Z.H. Han, B.W. Zan, K.S. Zhang, C. Gong, W.Z. Wang, Expert's experience-informed hierarchical kriging method for aerodynamic data modeling, in: Engineering Applications of Artificial Intelligence, Elsevier BV, 2024.
[55] A. Giunta, S. Wojtkiewicz, M. Eldred, Overview of modern design of experiments methods for computational simulations (invited), in: 41st Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2003.
[56] M.J. Sasena, P. Papalambros, P. Goovaerts, Exploration of metamodeling sampling criteria for constrained global optimization, Eng. Optim. 34 (3) (2002) 263–278, https://doi.org/10.1080/03052150211751.
[57] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Glob. Optim. 21 (2001) 345–383, https://doi.org/10.1023/A:1012771025575.
[58] J.D. Head, M.C. Zerner, A Broyden-Fletcher-Goldfarb-Shanno optimization procedure for molecular geometries, Chem. Phys. Lett. 122 (3) (1985) 264–270, https://doi.org/10.1016/0009-2614(85)80574-1.