33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS
2022
SWEDEN

AN EFFICIENT SOLUTION FOR AERODYNAMIC SHAPE OPTIMIZATION WITHIN THE MID-RANGE APPROXIMATION FRAMEWORK

Yu Zhang[1], Feng Qu[1], Junqiang Bai[1],

Dongsheng Jia[2], Elliot Karl Bontoft[2] & Vassili V. Toropov[2]

[1]Northwestern Polytechnical University, Xi’an 710072, PR China
[2]Queen Mary University of London, London E1 4NS, United Kingdom

Abstract

Metamodels offer a good solution for optimization problems with expensive experiments or simulations. With a large number of design variables, it is not easy to establish a precise metamodel throughout the whole design space for complex nonlinear problems, especially for a high-fidelity CFD analysis. To address this issue, the paper has presented an efficient solution for high-fidelity large-scale aerodynamic shape optimization problems based on several developments in the mid-range approximation method within a trust-region optimization framework. The trust-region strategy has been improved to contain more optimization states with a flexible and controllable performance to suit different types of problems. A metamodel assembly technique and its gradient-enhanced version are developed to further relax the requirements of computational costs in the mid-range approximation method. Its performance is discussed through a detailed comparison of metamodel performance by using the Vanderplaats scalable beam problem. The single wing of the Common Research Model is offered to the proposed method to conduct the aerodynamic shape optimization. With all constraints satisfied, the optimized configuration has a 4.85% improvement in wing drag performance. The results show that the proposed method could achieve the design goal successfully within a reasonable computational cost.

Keywords: mid-range approximation method; trust-region strategy; metamodel assembly; aerodynamic shape optimization; large-scale problems

1. Introduction

The prohibitive cost of the wind tunnel test makes more and more designers tend to apply Computational Fluid Dynamics (CFD) tools to analyze and optimize aerodynamic shapes in the main stage of the design process. The wind tunnel test currently is more to be used as a final check when an optimized aerodynamic shape is obtained. To reduce the chance that the final design may fail to satisfy design requirements under the wind tunnel test, high-fidelity methods should be used to produce more physically realistic designs. However, a new issue comes that significant computational resources are required for these high-fidelity methods. Furthermore, a detailed design with a sufficient number of design variables should be used to fully develop the potential of the design configurations. Considering the high computational cost of using high-fidelity models, it is quite difficult to explore the whole design space if optimizations have a large number of design variables, which attracts more attention to the efficient solutions for these problems.

Metamodels, also known as surrogate models, response surfaces, or approximation models, might be a good solution for the aforementioned issues. The use of metamodels would be the general process of creating a computationally inexpensive abstraction through the form of either an approximation or an interpolation of data with a well-fined set of inputs and outputs to replace the original expensive experiment or simulation [1]. It has been widely applied in different areas, including vehicle crashworthiness design [2], aircraft mission analysis [3], aerodynamic shape optimization [4], etc. But it is not easy to establish a precise metamodel throughout the whole design space for complex nonlinear problems, especially for a high-fidelity CFD analysis. What’s worse, with a large number of design variables, the difficulty of having a good metamodel increases. Although a direct solution is to provide more training points, the resulting computational consumption would be unacceptable, a situation that is referred to as “the curse of dimensionality”.

The Mid-range Approximation Method (MAM), which originated from the work of Haftka et al. [6] and was later developed by Toropov [7] and Toropov et al. [8], proves to be efficient to build better quality metamodels in large-scale problems with a reasonable number of training points [9,10]. Compared with the global-range approximation methods that build metamodels through the entire design space, the mid-range approximation methods establish metamodels in a subspace of the whole design space to reduce the difficulty of having a metamodel with good quality. The subspace of the whole design space is later referred to as the trust-region, which would be translated and scaled by the trust-region strategy during the design process. This method has been successfully used in turbomachinery design [11] and automotive structure design [12]. However, there are still few studies on aerodynamic shape optimizations of aircraft using the latest developments in the mid-range approximation methods.

In this paper, an efficient solution for aerodynamic shape optimization is presented based on several developments in the mid-range approximation method within a trust-region optimization framework. The trust-region strategy has been enhanced to include more optimization states for different types of problems. A Metamodel Assembly (MA) technique and its gradient-enhanced version are developed. Its performance is proved by a detailed comparison of metamodel performance by using the Vanderplaats scalable beam case [13]. Then an aerodynamic shape optimization of the Common Research Model (CRM) wing [14] is offered to demonstrate the proposed method.

2. Optimization Framework

2.1 Mid-range Approximation Method

MAM is an iterative optimization technique that could transform the original optimization problem into a sequence of approximate sub-optimization problems. This method can be seen as the application of the bilevel optimization techniques. The goal of its upper-level problem is to find a suitable trust-region that could build a good metamodel with a given number of training points and also include the optimum design. Then the lower-level problem is to solve the original problem in the current trust-region where all the physical experiments or simulations are replaced with metamodel predictions, which is referred to as the approximate sub-optimization problem.

A typical optimization process of MAM is shown in Figure 1. As the trust-region is moved on and changed, the optimization gradually approaches the optimum of the original problem, too. In every new trust-region, MAM needs to re-build the metamodel by using several new design points. However, it is normal to have some existing points inherited from previous iterations in the current trust-region. Using these existing points could save the computational cost in the evaluation of the physical simulations. Therefore, a non-collapsible randomized Design of Experiment (DoE) method [15] is selected to sample DoE points efficiently one by one with a reasonably uniform spread while taking into account the existing ones.

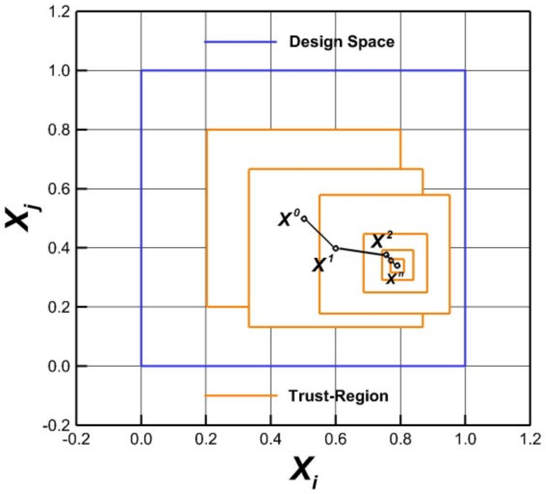


Figure 1 - Typical MAM optimization process

In this study, an improved trust-region strategy is offered to deal with the upper-level problem while a gradient-based optimizer, SLSQP (Sequential Least-Squares Quadratic Programming) algorithm [16], is used to solve the lower-level problem. To reduce the chance of falling into a local optimum, multiple approximate sub-optimizations could be conducted in every iteration of solving the upper-level problem. These sub-optimizations will start from several randomly initial points chosen in the current trust-region. Considering that all these sub-optimizations are solved based on metamodels, the resulting computational cost is relatively small.

2.2 Trust-Region Strategy

The choice of the trust-region strategy plays an important role in MAM for the performance and computational cost of the whole optimization. It provides a way to resolve the upper-level problem and determine the location and size of the trust-region in each iteration. In this paper, there are six input parameters for the considered trust-region strategy as follows:

- Metamodel Quality

It shows the discrepancy between the metamodels and physical simulations. Depending on the user-defined settings, the metamodel quality could be categorized into three types: “bad”, “good” and “precise”.

- Trust-Region Size

The trust-region size is the relative size of the current trust-region compared to the global design space. While the traditional trust-region strategy [17] divides it into two categories: “small” and “large”, the current strategy imports a new category “too small” to avoid the situation that all the sampled points have a similar performance that will bring ill-conditioned matrixes in the metamodel building stage.

- Optimum Location

This input parameter shows the location of the optimum point in the current trust-region. The traditional trust-region strategy distinguishes its state into two categories: “internal” and “external”. However, considering that the metamodel quality outside the current trust-region may not be as accurate as the one inside, this paper limits the optimum point to be selected only in the current trust-region. Hence, the current trust-region strategy discards the category “external” and subdivides the category “internal” into three types: “inside”, “near the boundary” and “at the boundary”. In this way, the input parameters could be generated based on more accurate information, and more detailed solutions could be provided for more detailed states.

- Feasibility

The feasibility of the optimum point is one of the important criteria for optimization convergence. The trust-region strategy also uses this parameter as an indicator to adjust the trust-region.

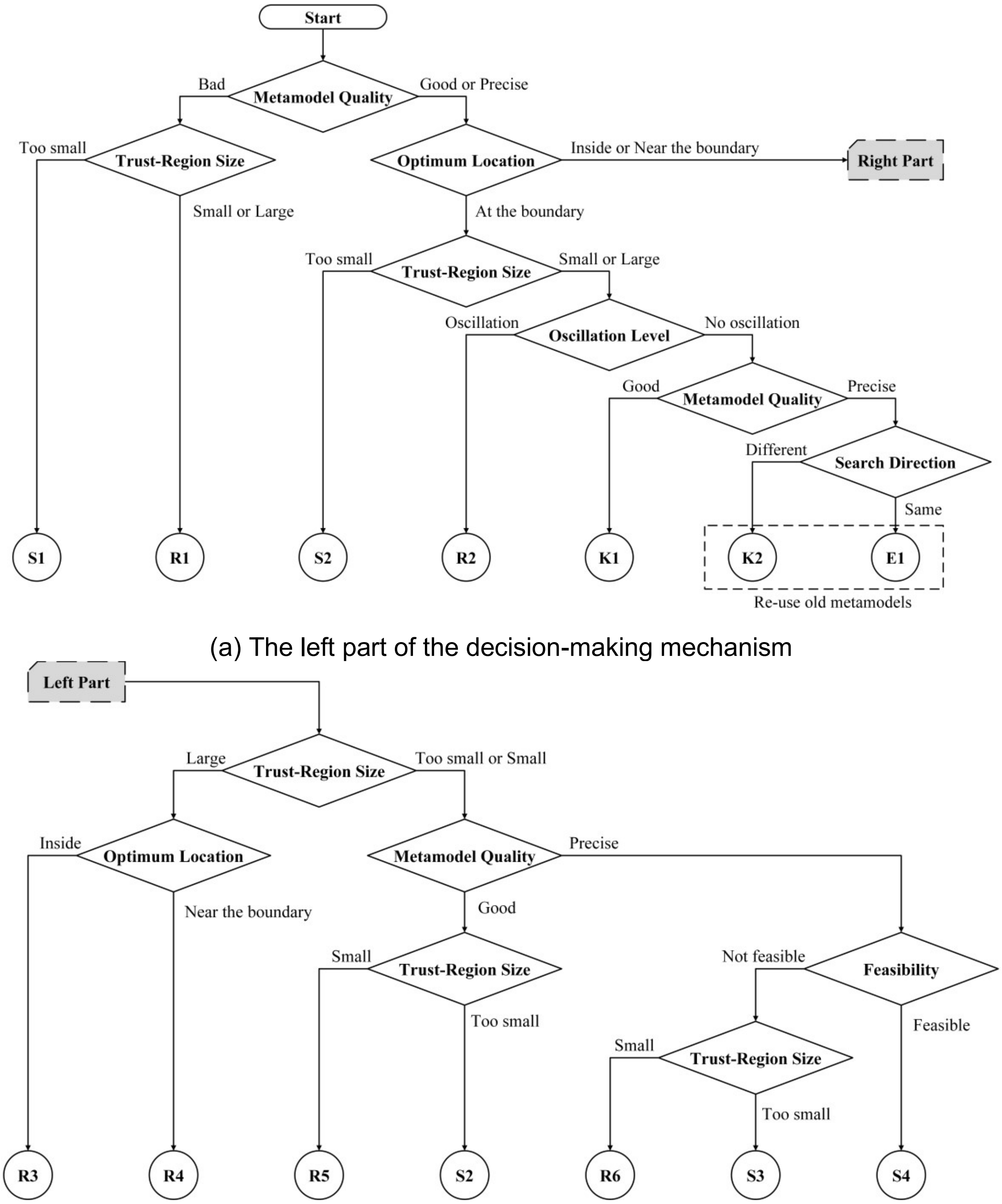
- Search Direction

The search direction derives from the movement history of the trust-region. If the optimum points are always found “at the boundary” and the trust-region keeps moving in the same direction, it is necessary to enlarge the trust-region to contain more design space and improve the optimization efficiency.

- Oscillation Level

This parameter is to check if the trust-region, or the optimization, oscillates around certain points. If it is, a special treatment is required to adjust the trust-region while ending this oscillation.

With these functions as input parameters, the trust-region strategy will set the optimum location as the centre of the new trust-region and apply the decision-making mechanism shown in Figure 2 to determine the new trust-region size. This decision-making mechanism has defined 13 optimization states according to the combination of the above six parameters, including 4 stop states (S1-S4), 6 reduction states (R1-R6), 1 enlargement state (E1), and 2 keeping states (K1, K2). Different optimization states have different solutions to modify the trust-region. Normally, the stop states mean the optimization should be stopped. The reduction states mean the strategy should reduce the trust-region size in the next iteration while the enlargement state should increase it. Similarly, the keeping states indicate the trust-region size should remain unchanged. With the subdivision and re-classification of part parameters, the improved trust-region strategy has contained more optimization states with a flexible and controllable performance to suit different types of problems.



(a) The left part of the decision-making mechanism

(b) The right part of the decision-making mechanism

Figure 2 - Decision-making mechanism of the trust-region strategy

2.3 Metamodel Assembly Technique

Although the mid-range approximation method makes it possible to employ metamodel techniques in large-scale problems, the selection of metamodel techniques will greatly affect the optimization performance. As the number of design variables and design responses grows, the computational cost of the metamodel building and evaluation becomes a non-negligible part of the whole optimization. Especially for the mid-range approximation method, metamodels need to be re-built and evaluated in every iteration since the trust-region is moving as the optimization goes on, which raises higher requirements for the efficiency of the metamodel techniques.

Considering these issues, this work has developed a metamodel assembly technique and its gradient-enhanced version originating from the work in reference [18]. By using the MA method, multiple metamodels could be assembled into one single metamodel to combine their strengths properly. Such an approach has been widely studied in different applications with different metamodels, including polynomial regression (PR), radial basis function (RBF), Kriging (KRG), etc [19-21]. However, building and evaluation of these classical metamodels may involve a lot of matrix computations, whose efficiency would be greatly worse in large-scale problems. What's worse, some metamodel methods require a pre-optimization for their tuning parameters, known as hyperparameter optimization [22,23], making it impossible to be used in large-scale problems.

In this way, the paper has selected five linear or intrinsically linear functions [24] as basic metamodels as follows:

$$\begin{aligned}
\varphi_1(\mathbf{x},\mathbf{a}_1) &= a_{1,0}+\sum_{j=1}^{n}a_{1,j}\times x_j\\
\varphi_2(\mathbf{x},\mathbf{a}_2) &= a_{2,0}+\sum_{j=1}^{n}a_{2,j}\times \frac{1}{x_j}\\
\varphi_3(\mathbf{x},\mathbf{a}_3) &= a_{3,0}+\sum_{j=1}^{n}a_{3,j}\times x_j^2\\
\varphi_4(\mathbf{x},\mathbf{a}_4) &= a_{4,0}+\sum_{j=1}^{n}a_{4,j}\times \frac{1}{x_j^2}\\
\varphi_5(\mathbf{x},\mathbf{a}_5) &= a_{5,0}\times\prod_{j=1}^{n}x_j^{a_{5,j}}
\end{aligned} \tag{1}$$

where φ is the basic metamodel, $\mathbf{x}$ is the design variable vector from sampling points, n is the number of design variables, $\mathbf{x}=[x_1,\cdots,x_n]$, $\mathbf{a}_l$ is the tunning vector in the l^{th} metamodel, $\mathbf{a}_l=[a_{l,0},a_{l,1},\cdots,a_{l,n}]$ and $l=1,2,\cdots,5$. The intrinsically linear function is nonlinear but could be converted to a linear function. Every basic metamodel has $n+1$ tunning parameters, which are solved by the linear regression method as shown in the following formulation:

$$min\ \sum_{p=1}^{P}w_p\times\left[F(\mathbf{x}_p)-\varphi_l(\mathbf{x}_p,\mathbf{a}_l)\right]^2 \tag{2}$$

where P is the number of training points, p is the index of the selected training point, F is the response function from the physical experiments or simulations, and w_p is the weight factor of the corresponding point x_p. If the gradient information is available, we could incorporate it to improve the metamodel quality with the following equation:

$$min\ \sum_{p=1}^{P}w_p\times\left\{\left[F(\mathbf{x}_p)-\varphi_l(\mathbf{x}_p,\mathbf{a}_l)\right]^2+\sum_{j=1}^{n}\gamma\times\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_j}-\frac{\partial\varphi_l(\mathbf{x}_p,\mathbf{a}_l)}{\partial x_j}\right]^2\right\} \tag{3}$$

where γ is a parameter to show how important the gradient information is compared with the response information. In this study, γ is set to 0.5. Notice that by using the linear regression method, the required minimum number of training points is $n+1$ if the gradient information is unavailable and 1 if the gradient information is provided, which allows this method uses as few samples as possible to build metamodels.

When we finish the building of basic metamodels, the next step is to combine them using the following formulation:

$$F(\mathbf{x},\mathbf{b})=\sum_{l=1}^{nf}b_l\times\varphi_l(\mathbf{x},\mathbf{a}_l) \tag{4}$$

where F is the combined metamodel to replace the original function F, $\mathbf{b}$ is the tunning vector of the metamodel assembly, nf is the number of the basic metamodels which is 5 in this work, $\mathbf{b}=[b_1,b_2,\cdots,b_{nf}]$. Similarly, the tunning vector $\mathbf{b}$ is solved by the linear regression method using the following formulation

$$min\ \sum_{p=1}^{P}w_p\times\left[F(\mathbf{x}_p)-F(\mathbf{x}_p,\mathbf{b})\right]^2 \tag{5}$$

If the gradient information is offered, the following extension could be used:

$$min\ \sum_{p=1}^{P} w_p \times \left\{ \left[F\left(\mathbf{x}_p\right)-F\left(\mathbf{x}_p,\mathbf{b}\right)\right]^2+\sum_{j=1}^{n}\gamma\times\left[\frac{\partial F\left(\mathbf{x}_p\right)}{\partial x_j}-\frac{\partial F\left(\mathbf{x}_p,\mathbf{b}\right)}{\partial x_j}\right]^2\right\} \tag{6}$$

When the tunning vectors $\mathbf{a}_l$ and $\mathbf{b}$ are obtained, equation (4) is used to predict the response values for different test points.

Compared with classical metamodel techniques like RBF and KRG, the MA technique proposed in this paper consists of several linear or intrinsically linear functions that are simple enough to be built and evaluated efficiently, which could relax the efficiency issues of the metamodel technique in large-scale problems. This is also proved by a detailed comparison of metamodel performance shown in **Section 3.1**. Besides, parallel computing technology has been implanted in the proposed method to build and evaluate metamodels for different responses simultaneously.

3. Benchmark and Application

3.1 Vanderplaats scalable beam

A benchmark test case known as Vanderplaats scalable beam [13] is studied in this section to demonstrate the proposed method first. As shown in Figure 3, the research subject is a cantilevered beam that consists of *S* segments with rectangular cross-sections. The number *S* can be chosen arbitrarily.

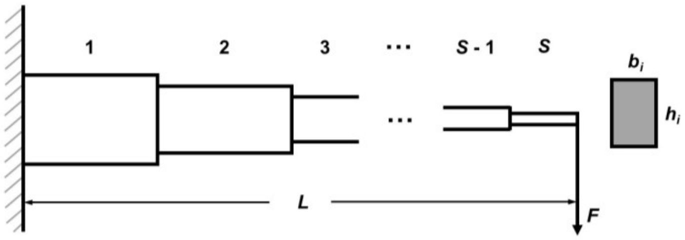


Figure 3 - Vanderplaats scalable beam

The detailed optimization model is shown in Table 1. In this optimization problem, the design objective is to minimize the beam volume *V*. The stress σ_i and aspect ratio h_i / b_i of every segment are considered as the design constraints. The tip deflection y_S due to the external load F = 5×10^4 *N* is seen as a global constraint. The design variables are the widths b_i and heights h_i of these rectangular cross-sections. The total length *L* is 500 *cm* which means the length of every segment l_i is 500 / *S cm*. All the responses and their gradient information could be calculated by analytical functions given in reference [13].

Table 1 - Optimization model of Vanderplaats scalable beam

	Objective / Constraint / Variable	Description	Quantity
minimize	$V=\sum_{i=1}^{S} b_i \times h_i \times l_i$	Beam volume	1
with respect to	1 *cm* ≤ b_i ≤ 10 *cm*	Widths	*S*
	5 *cm* ≤ h_i ≤ 100 *cm*	Heights	*S*
		Total number of design variables	2 × *S*
subject to	σ_i ≤ 14000 *N / cm*^{2}	Stress constraints	*S*
	h_i / b_i ≤ 20	Aspect ratio constraints	*S*
	y_S ≤ 2.5 *cm*	Tip deflection constraint	1
		Total number of design constraints	2 × *S* + 1

Here we suppose that the considered problem has 256 segments, which means there are 1 design objective, 512 design variables, and 513 design constraints. That could be seen as a typical large-scale problem discussed in this paper. Two MAM optimization cases have been conducted. The first one doesn't use gradient information which is labelled as MAM and the second one does utilize gradients which is labelled as GEMAM (Gradient-Enhanced MAM). In each iteration of the upper-level problem, the MAM case samples 1024 training points to build metamodels while the GEMAM case samples 6 training points. And three approximate sub-optimizations have been conducted in every iteration to reduce the possibility of local optimal. The initial trust-region size in both cases is set to 25 %. To verify the optimization performance of the proposed method, a gradient-based optimization case using the SLSQP algorithm is also offered as a reference. The desired accuracy of SLSQP is 1 $\times10^{-6}$.

The optimization results from three cases are compared in Table 2 and their optimization histories are plotted in Figure 4. For this problem, Both the MAM case and GEMAM case arrive at a solution that is quite similar to the gradient-based method, with relative errors of less than 0.05%. All design constraints have been satisfied. For the MAM case, the proposed method has successfully obtained a good solution at an acceptable cost of 19549 response evaluations. The other classical gradient-free methods like the Genetic Algorithm (GA) [27] or Simulated Annealing Algorithm (SAA) [28] might need several times or even tens of times the same computational resources to find a similar solution for this problem that has 512 design variables. As for the GEMAM case, the gradient information is used to build metamodels, which could improve the metamodel quality and reduce the required number of DoE points. Compared with the SLSQP case, a slightly better solution has been achieved with a lower computational cost. Notice that for both MAM and GEMAM cases, every iteration plotted in Figure 4 contains three approximate sub-optimizations starting from random initial points, which make the optimization less likely to be trapped in local optimal. But every optimized point from approximate sub-optimizations needs to be re-evaluated by physical simulation solvers. If we use a smaller number of approximate sub-optimizations in every iteration, the overall computational cost used by GEMAM could be further reduced. These results prove that the proposed method can provide an efficient solution for large-scale problems with good accuracy.

Table 2 - Optimization results for Vanderplaats scalable beam problem with 256 segments

Method	Objective	Relative Error	Iterations	Response Evaluations	Gradient Evaluations
SLSQP	63691.58	-----	215	217	216
MAM	63683.73	0.0123%	19	19549	0
GEMAM	63668.54	0.0362%	19	172	172

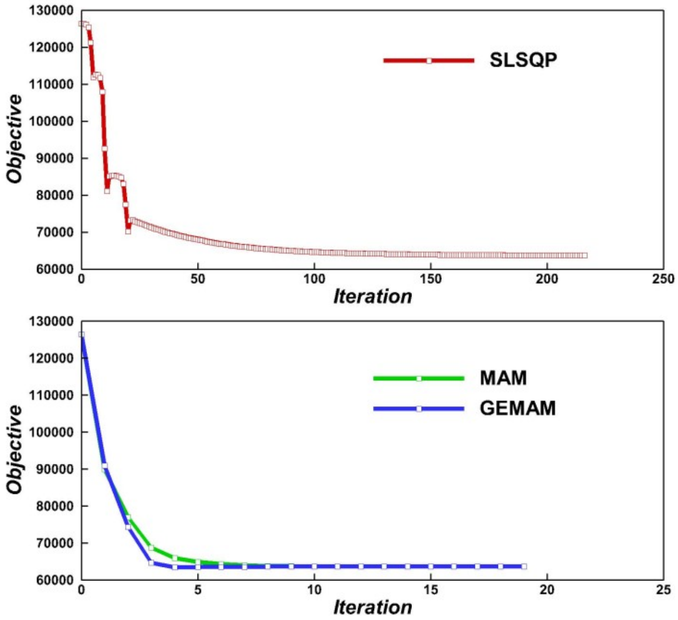


Figure 4 - Optimization histories of Vanderplaats scalable beam problem with 256 segments

Next, using the same problem, we have compared the performance of different metamodels. The considered metamodels include MA, KRG and MLS (Moving Least-Squares) [29]. MA is the metamodel assembly technique developed in **Section 2.3**. KRG is a popular and typical approximation technique in classical metamodels. And MLS is one of the most frequently used choices of metamodel technique in studies using the mid-range approximation method [10,11,30].

In this study, 11 cases are extracted in which the number of segments ranges from 10 to 100. That means the range of the number of design variables is between 20 and 200, and the range of the number of responses that should be replaced by metamodels is between 22 and 202. The building time, evaluation time and prediction error of metamodels are extracted as indicators to demonstrate metamodel performance. Considering the existence of multiple responses, the average values of these indicators among all the responses are used for the final comparison. Furthermore, to eliminate the influence of randomness in the selected DoE method, we have repeated this process including building and evaluation 50 times for all 11 cases and then, similarly, their average values are extracted.

For all 11 test cases, two trials have been conducted. The first trial builds metamodels only with responses and the second trial builds metamodels with responses and their gradients. For the first trial, the number of training points is set to twice the number of design variables. And for the second trial, the number of training points is fixed at 6. The corresponding labels of these metamodels in the second trial are named GEMA (Gradient-Enhanced MA), GEKRG (Gradient-Enhanced KRG) and GEMLS (Gradient-Enhanced MLS), respectively. All these trials in all cases are tested in a single-core computational environment.

Figure 5 and Figure 6 give the results of the metamodel performance comparison in the first and second trials. In both trials, the MLS method has the least building time and the largest evaluation time while the KRG model has the largest building time and the second larger evaluation time. The MA technique has the second larger building time and the least evaluation time. Considering that all metamodels will be re-built once and evaluated several times at every MAM iteration, the evaluation time of metamodels should be the most important part to affect the overall computational cost. Therefore, the MA technique which has the least evaluation time should be a suitable choice.

Besides, as the number of design variables increases, the MA technique is the least affected one. When the number of design variables increases by one order of magnitude (from 20 to 200), the building time of MA increases by nearly two orders of magnitude in both trials and its evaluation time increases by just approximately one order of magnitude. But for KRG trials, its building and evaluation time rise almost 10000 times and 1000 times. In GEKRG trials, its building and evaluation time both increase 100 times. Following this trend, the computational cost of the building and evaluation of KRG metamodels would increase rapidly to be unacceptable for the large-scale problems with hundreds of design variables. As for the MLS method, although its building time could be ignored compared with other methods, the issue in its evaluation time is more serious than the KRG method.

Finally, the MA technique also has the best performance in the prediction error. However, this might only suit the current problem since KRG and MLS both need hyperparameter optimizations in advance to get a good metamodel quality. In these trials, we disable the hyperparameter optimizations and use a fixed set of hyperparameters, which leads to poor performance on the prediction error of KRG and MLS. The hyperparameter optimizations need to re-build and re-evaluate the metamodels several times. Considering their rapid increase in the building and evaluation time, the KRG and MLS metamodels using hyperparameters optimizations are not suitable for large-scale problems.

Based on the discussion above, the developed MA method and its gradient-enhanced version are proved to be a useful and efficient metamodel technique that could be used in the mid-range approximation method for large-scale problems.

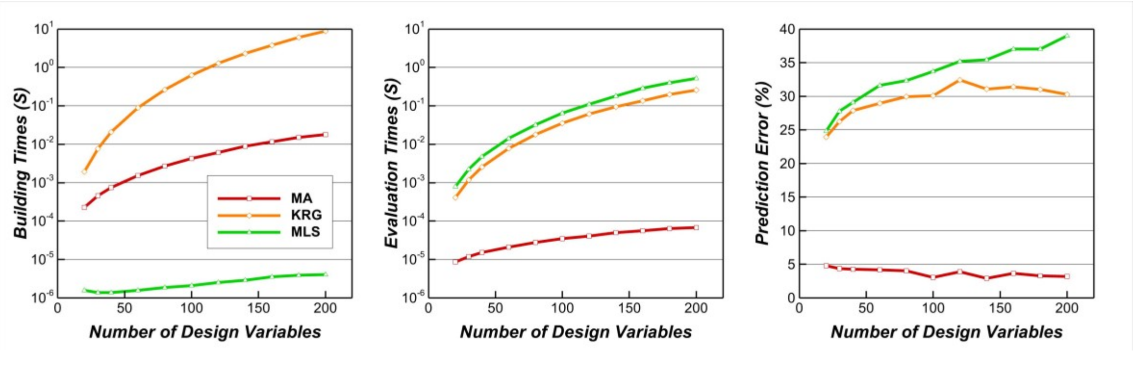


Figure 5 - Metamodel performance comparison without gradient information

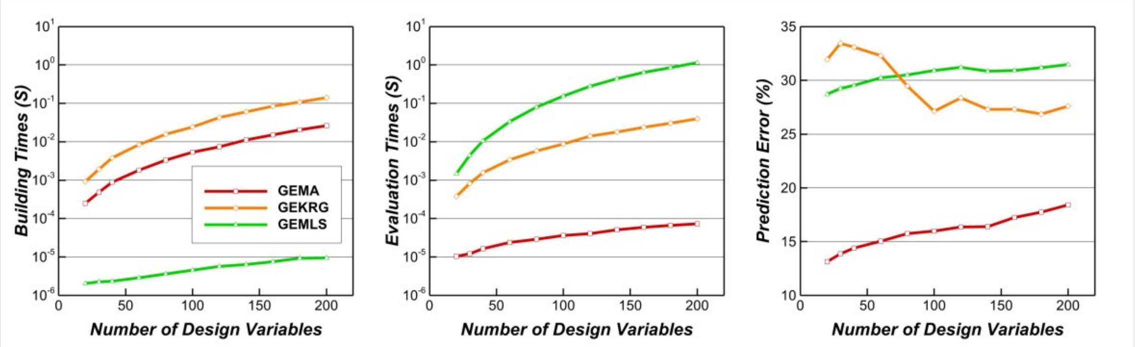


Figure 6 - Metamodel performance comparison with gradient information

3.2 Aerodynamic shape optimization

In this section, the proposed method is applied to a wing aerodynamic shape optimization problem. As shown in Figure 7, the wing is extracted from the CRM wing-body geometry and its leading edge (LE) at the wing root section is set as the origin of coordinates. The wing span is 26.327 *m*. The reference chord and reference area are 7.005 *m* and 167.198 m^2, respectively. The orange dot in the figure is the reference point used in this problem, which is (8.460, 0.000, 0.054) *m*. The flight design condition is shown as follows:

$$Ma = 0.85, Re = 40 \times 10^6, C_L = 0.50$$

The left part in Figure 8 gives the wing CFD grid. This multi-block structure grid has 3.396 million cells. All cells near the wall have been adjusted to make the maximum dimensionless wall distance y^+ less than or equal to 1. Using this grid, one CFD evaluation based on Reynolds-Averaged Navier-Stokes (RANS) equations might take hours. To minimize the number of CFD evaluations required in the optimization, the gradient information should be applied to build metamodels in MAM. Therefore, a RANS-based CFD solver using Spalart-Allmaras (SA) turbulence model, namely ADflow [31], is selected to analyze the aerodynamic performance of design configurations. ADflow has been differentiated with the adjoint method and can finish gradient computations efficiently.

A Free Form Deformation (FFD) method [25] is applied to parameterize the wing shape. The FFD method puts the original object into a flexible control box, like the right part shown in Figure 8, and then builds a mathematical mapping relationship between object and box. Through this relationship, one can modify the wing surface arbitrarily by adjusting the position of the control points in the control box. The position of these control points could be treated as the design variables in the optimization. When a new perturbed shape is generated, a mesh deformation algorithm using the inverse distance weighting interpolation method [26] is used to propagate the surface changes to volume grids.

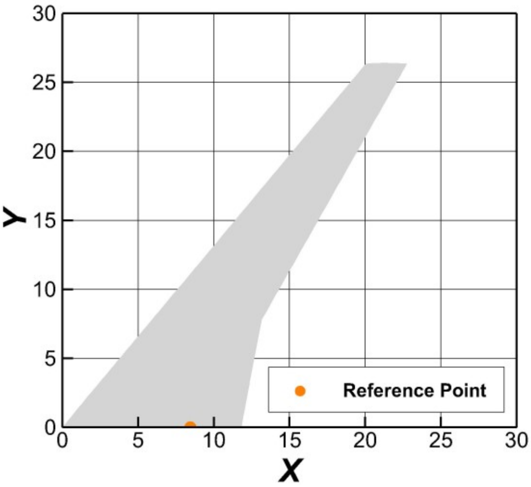


Figure 7 - CRM wing geometry

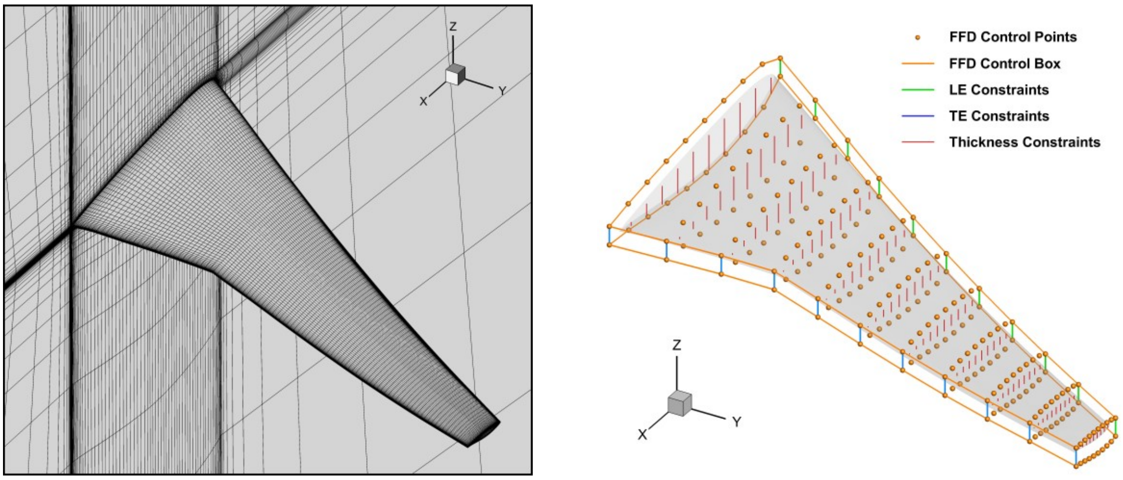


Figure 8 - CFD grid (left) and FFD control box (right)

Table 3 illustrates the detailed optimization model of wing aerodynamic shape optimization. The considered problem has 1 design objective, 232 design variables and 135 design constraints. The optimization tries to seek the minimum drag coefficient C_D subject to a lift constraint (C_L = 0.50) and a pitching moment constraint ($C_M \geq -0.19$). The lift constraint ensures that the computational condition will not deviate from the required design condition during the optimization. To satisfy this constraint, the angle of attack (*AoA*) is included in the optimization as a design variable. The pitching moment constraint limits the pitching moment to be greater than a given value in case the trim drag coefficient would deteriorate in the following design of the complete aircraft.

Table 3 - Wing aerodynamic shape optimization model

	Objective / Constraint / Variable	Description	Quantity
minimize	C_D	Drag coefficient	1
with respect to	*AoA*	Angle of Attack	1
	X_{FFD}	FFD control points	220
	X_{twist}	Wing twist	11
		Total number of design variables	232
subject to	$C_L = 0.50$	Lift constraint	1
	$C_M \geq -0.19$	Pitching moment constraint	1
	$X_{twist,\ root} = 0$	Wing root constraint	1
	$X_{LE,\ Upper} = -X_{LE,\ Lower}$	LE constraints	11
	$X_{TE,\ Upper} = -X_{TE,\ Lower}$	TE constraints	11
	$t \geq t_0$	Thickness constraints	110
		Total number of design constraints	135

The geometry design variables and constraints are shown in the right part of Figure 8. The control box consists of 11 control sections distributed along with wing span-wise direction (*Y* axis). For each control section, there are 20 control points, shown as orange spheres in the figure, 10 on the upper surface and 10 on the lower surface. By using the FFD method, the modification of each control point could efficiently change the wing surface nearby. During the optimization, the displacement of these control points in the *Z* axis is treated as local shape variables X_{FFD}. And the rotations of these control sections around the *Y* axis are seen as the global shape variables X_{twist} to control the wing span-wise twist distribution. The rotation axis of each control section is located on the wing leading edge.

Then several geometry constraints have been imported according to some engineering considerations. Firstly, the twist at the root section is fixed to keep the angle of wing incidence unchanged. Then, we find that if the perturbation of X_{FFD} occurs in the leading edge or trailing edge (TE), the wing span-wise distribution could also be slightly affected. To reduce the influence of interaction between X_{FFD} and X_{twist}, one LE constraint and one TE constraint have been put on every control section, shown as the green and blue lines in Figure 8. These LE / TE constraints limit the control points at these positions to move the same distance in the opposite direction. Then X_{twist} could be the only factor to influence the wing span-wise twist distribution. Finally, in order not to reduce the wing volume, 110 wing thickness constraints have been evenly distributed in the wing plane, see the red thick line in Figure 8. These thickness constraints restrict the thickness at the corresponding position to be equal to or larger than their initial values during the optimization.

The same MAM settings have been used in this problem. The initial size of the trust-region is set to 25%. In each iteration of the upper-level problem, we would sample 6 training points to build GEMA. To reduce the probability of falling into a local optimum, three approximate sub-optimizations have been conducted in every iteration. The whole optimization works in a High-Performance Computing (HPC) environment with 144 cores [32].

As shown in Figure 9, the proposed method successfully finished the optimization in 86 iterations with a computational cost of 481 response and gradient evaluations. The consumed number of response and gradient evaluations is approximately 2.07 times the number of design variables. The detailed aerodynamic coefficients between the baseline and optimized configurations are listed in Table 4. All constraints including the lift and pitching moment constraints have been satisfied. The design objective, drag coefficient $C_{D\ counts}$, decreases by 8.32 counts, which is a 4.85% improvement in wing drag performance. In this paper, all drag coefficients C_D are given in counts. $C_{D\ counts} = C_D \times 10^{-4}$.

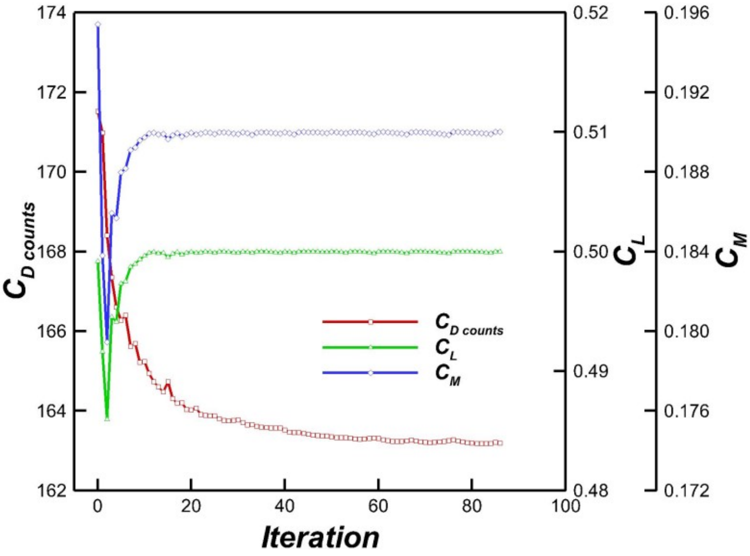


Figure 9 - Convergence history of wing aerodynamic shape optimization

Table 4 - Aerodynamic performance at the design condition

Configuration	*AoA* (º)	C_L	$C_{D\ counts}$	C_M	*L / D*
Baseline	1.898	0.50	171.52	-0.1954	29.15
Optimized	2.070	0.50	163.20	-0.1900	30.64

To give an understanding of how the proposed method works, we plot the optimization history of trust-region size and metamodel quality as shown in Figure 10. These two functions are key indicators of the trust-region strategy. In the first 20 iterations of optimization, the trust-region size gradually reduced with a more and more accurate metamodel. MAM has initially identified the trust-region including the optimum. The lift coefficient and the pitching moment coefficient approached the boundary of constraints as shown in Figure 9. Then the decreasing speed in the drag coefficient slowed down. To further reduce the drag coefficient, the optimization turned to give a thoroughly searching with mild steps. The trust-region would be slightly reduced in this stage to maintain the metamodel quality at a good level. Finally, the optimization stopped with a normal convergence where prediction errors are below 10^{-5} and optimal designs from approximate sub-optimizations are nearly identical to the actual optimum design.

Figure 11 shows a comparison of aerodynamic performance between the baseline and optimized configuration. The contours of pressure coefficient C_P on the upper wing surface are given first. The area of pressure concentration in the middle wing where the shock exists has greatly reduced. The optimized design has a smooth and nearly parallel pressure distribution. The following C_P distributions and airfoil shapes at six span-wise sections also prove the good performance of the optimized design.

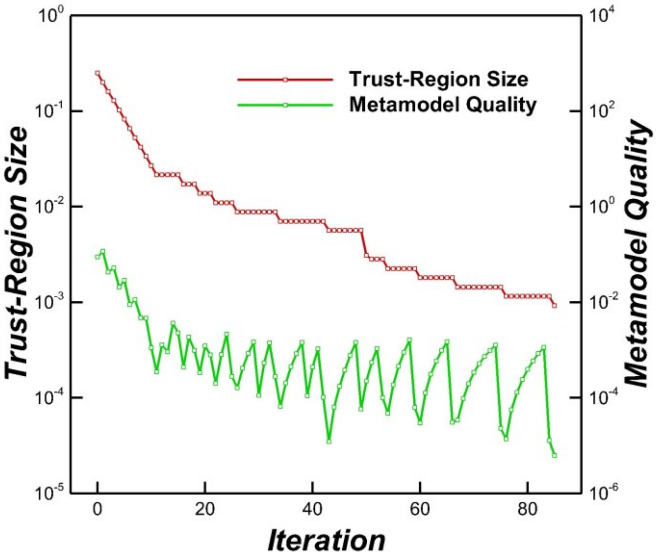


Figure 10 - Optimization history of trust-region size and metamodel quality

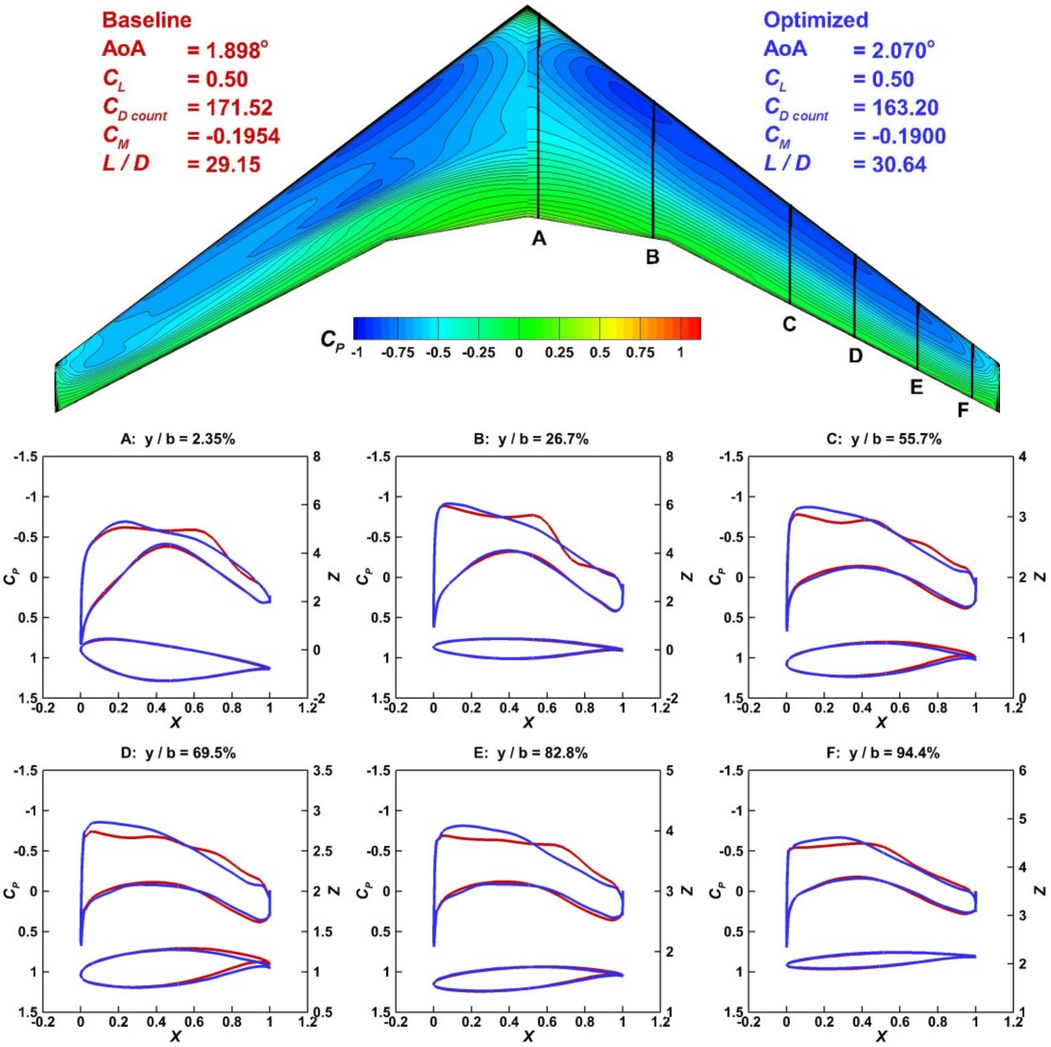


Figure 11 - Comparison of aerodynamic performance between the baseline and optimized configuration

4. Conclusion

This paper has presented an efficient solution for high-fidelity large-scale aerodynamic shape optimization problems based on several developments in the mid-range approximation method within a trust-region optimization framework. The mid-range approximation method is a bilevel optimization technique that converts the original optimization problem into a sequence of approximate sub-optimization problems. It provides a way for designers to utilize metamodel techniques in high-fidelity large-scale problems.

The trust-region strategy has been improved to contain more optimization states with a flexible and controllable performance to suit different types of problems. It defines 13 optimization states based on the combination of six indicators. Different optimization states give different solutions to upper-level problems. Then the upper-level problem could be solved efficiently and robustly.

The metamodel assembly technique and its gradient-enhanced version have been developed to further improve the performance of the mid-range approximation method. Compared with part classical metamodels like KRG and MLS, the proposed metamodel assembly technique could efficiently build and evaluate metamodels and have relatively good prediction quality in the selected trust-region without extra hyperparameter optimizations. This has been confirmed by multiple trials in the Vanderplaats scalable beam problem with different numbers of segments from 10 to 100.

The proposed method has been applied to a benchmark case and a wing aerodynamic shape optimization problem. The benchmark case is a Vanderplaats scalable beam problem with 256 segments, which has 1 design objective, 512 design variables and 513 design constraints. The proposed method has successfully found the optimum solution at a reasonable computational cost. With the gradient-enhanced metamodel assembly technique, the computational consumption is even better than the selected gradient-based method SLSQP. The aerodynamic shape optimization of the CRM wing consists of 1 design objective, 232 design variables and 135 design constraints. The final solution has a 4.85% improvement in wing drag performance. The shock region has been greatly reduced. These results demonstrate the effectiveness of the proposed method in high-fidelity large-scale aerodynamic shape optimization problems.

5. Contact Author Email Address

For further communication, the corresponding authors' email addresses are shown as follows:

Yu Zhang, mailto: guardmdo@mail.nwpu.edu.cn

Dongsheng Jia, mailto: d.jia@qmul.ac.uk

6. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third-party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Viana F A C, Simpson T W, Balabanov V, and Toropov V V. Metamodeling in multidisciplinary design optimization: how far have we really come?. *AIAA Journal*, Vol. 52, No. 4, pp 670-690, 2014. https://doi.org/10.2514/1.J052375.

[2] Yin H, Fang H, Wen G, Gutowski M, and Xiao Y. On the ensemble of metamodels with multiple regional optimized weight factors. *Structural and Multidisciplinary Optimization*, Vol. 58, No. 1, pp 245-263, 2018. https://doi.org/10.1007/s00158-017-1891-1.

[3] Liem R P, Mader C A, and Martins J R R A. Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis. *Aerospace Science and Technology*, Vol. 43, pp 126-151, 2015. https://doi.org/10.1016/j.ast.2015.02.019.

[4] Han Z, Xu C, Zhang L, Zhang Y, Zhang K, and Song W. Efficient aerodynamic shape optimization using variable-fidelity surrogate models and multilevel computational grids. *Chinese Journal of Aeronautics*, Vol. 33, No. 1, pp 31-47, 2020. https://doi.org/10.1016/j.cja.2019.05.001.

[5] Liu D Z, and Toropov V V. Implementation of discrete capability into the enhanced multipoint approximation method for solving mixed integer-continuous optimization problems. *International Journal for Computational Methods in Engineering Science and Mechanics*, Vol. 17, No. 1, pp 22-35, 2016. https://doi.org/10.1080/15502287.2016.1139013.

[6] Haftka R T, Nachlas J A, Watson L T, Rizzo T, and Desai R. Two-point constraint approximation in structural optimization. *Computer Methods in Applied Mechanics and Engineering*, Vol. 60, No. 3, pp 289-301, 1987. https://doi.org/10.1016/0045-7825(87)90136-8.

[7] Toropov V V. Simulation approach to structural optimization. *Structural Optimization*, Vol. 1, No. 1, pp 37-46, 1989. https://doi.org/10.1007/BF01743808.

[8] Toropov V V, Filatov A A, and Polynkin A A. Multiparameter structural optimization using FEM and multipoint explicit approximations. *Structural Optimization*, Vol. 6, No. 1, pp 7-14, 1993. https://doi.org/10.1007/BF01743169.

[9] Polynkin A A, and Toropov V V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Structural and Multidisciplinary Optimization*, Vol. 45, No. 4, pp 515-527, 2012. https://doi.org/10.1007/s00158-011-0692-1.

[10] Taherkhani A R, Gilkeson C, Gaskell P, Hewson R, Toropov V V, Rezaienia A, Thompson H. Aerodynamic CFD based optimization of police car using bezier curves. *SAE International Journal of Materials and Manufacturing*, Vol. 10, No. 2, pp 85-93, 2017. https://doi.org/10.4271/2017-01-9450.

[11] Caloni S, Shahpar S, and Toropov V V. Multi-disciplinary design optimisation of the cooled squealer tip for high pressure turbines. *Aerospace*, Vol. 5, No. 4, pp 116-136, 2018. https://doi.org/10.3390/aerospace5040116.

[12] Mortished C, Ollar J, Benzie P, Jones R, Sienz J, and Toropov V V. Multidisciplinary optimisation of an automotive body-in-white structure using crushable frame springs and sub space metamodels in trust-regions. *Advances in Structural and Multidisciplinary Optimization*, pp 1572-1584, 2018. https://doi.org/10.1007/978-3-319-67988-4_118.

[13] Vanderplaats G N. Numerical optimization techniques for engineering design: with applications. *Vanderplaats Research & Development, Inc*, pp 643-648, 1984.

[14] Vassberg J C, DeHaan M A, Rivers S M, and Wahls R A. Development of a Common Research Model for applied CFD validation Studies. *26th AIAA Applied Aerodynamics Conference*, Honolulu, Hawaii, AIAA 2008-6919, 2008.

[15] Korolev Y M, Toropov V V, and Shahpar S. Large-scale CFD optimization based on the FFD parametrization using the multipoint approximation method in an HPC environment. *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX, AIAA 2015-3234, 2015.

[16] Kraft D. A software package for sequential quadratic programming. *Technical Report DFVLR-FB*, Vol. 88, No. 28, pp 1-33, 1988.

[17] Keulen F Van, and Toropov V V. New developments in structural optimization using adaptive mesh refinement and multipoint approximations. *Engineering Optimization*, Vol. 29, No. 1-4, pp 217-234, 1997. https://doi.org/10.1080/03052159708940994.

[18] Polynkin A A, and Toropov V V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Structural and Multidisciplinary Optimization*, Vol. 45, No. 4, pp 515-527, 2012. https://doi.org/10.1007/s00158-011-0692-1.

[19] Viana F A C, and Haftka R. Using Multiple Surrogates for Metamodeling. *Proceedings of the 7th ASMO-UK/ISSMO International conference on engineering design optimization*, Bath, UK, 2008.

[20] Han Z H, Görtz S, and Zimmermann R. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, Vol. 25, No. 1, pp 177-189, 2013. https://doi.org/10.1016/j.ast.2012.01.006.

[21] Yin H, Fang H, Wen G, Gutowski M, and Xiao Y. On the ensemble of metamodels with multiple regional optimized weight factors. *Structural and Multidisciplinary Optimization*, Vol. 58, No. 1, pp 245-263, 2018. https://doi.org/10.1007/s00158-017-1891-1.

[22] Strijov V, and Weber G W. Nonlinear regression model generation using hyperparameter optimization. *Computers and Mathematics with Applications*, Vol. 60, No. 4, pp 981-988, 2010. https://doi.org/10.1016/j.camwa.2010.03.021.

[23] Ollar J, Mortished C, Jones R, Sienz J, and Toropov V V. Gradient based hyper-parameter optimisation for well conditioned kriging metamodels. *Structural and Multidisciplinary Optimization*, Vol. 55, No. 6, pp 2029-2044, 2017. https://doi.org/10.1007/s00158-016-1626-8.

[24] Box G E P, and Draper N R. Empirical model-building and response surfaces. *Journal of the American Statistical Association*, Vol. 83, No. 402, pp 569-570, 1988. https://doi.org/10.2307/2288890.

[25] He X, Li JC, Mader C A, Yildirim A, and Martins J R R A. Robust aerodynamic shape optimization - from a circle to an airfoil. *Aerospace Science and Technology*, Vol. 87, pp 48-61, 2019. https://doi.org/10.1016/j.ast.2019.01.051.

[26] Witteveen J A S, and Bijl H. Explicit mesh deformation using inverse distance weighting interpolation. *19th AIAA Computational Fluid Dynamics Conference*, San Antonio, Texas, AIAA 2009-3996, 2009.

[27] Deb K, Pratap A, Agarwal S, and Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp 182-197, 2002. https://doi.org/10.1109/4235.996017.

[28] Welsh D J A. Simulated annealing: theory and applications. *Bulletin of the London Mathematical Society*, Vol. 21, No. 2, pp 204-205, 1989. https://doi.org/10.1112/blms/21.2.204b.

[29] Lancaster P, and Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation*, Vol. 37, No. 155, pp 141-158, 1981. https://doi.org/10.2307/2007507.

[30] Gilkeson C A, Toropov V V, Thompson H M, Wilson M C T, Foxley N A, and Gaskell P H. Dealing with numerical noise in CFD-based design optimization. *Computers and Fluids*, Vol. 94, pp 84-97, 2014.

https://doi.org/10.1016/j.compfluid.2014.02.004.

[31] Mader C A, Kenway G K W, Yildirim A, and Martins J R R A. ADflow: an open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, Vol. 17, No. 9, pp 508-527, 2020. https://doi.org/10.2514/1.I010796.

[32] King T, Butcher S, and Zalewski L. Apocrita - High Performance Computing cluster for Queen Mary University of London. *Queen Mary University of London, Technical Report*, pp 1-2, 2017. http://doi.org/10.5281/zenodo.438045.