Efficient aerodynamic shape optimization through free-form mesh morphing and reduced order CFD modeling

S F Peng[1], Z P Liu[2], C Y Zhang[3]*

1. School of Industrial Design, Guangzhou Academy of Fine Arts, China
2. State Key Laboratory of Nuclear Power Safety Monitoring Technology and Equipment, China Nuclear Power Engineering Co. Ltd., China
3. Sino-French Institute of Nuclear Engineering & Technology, Sun Yat-Sen University, China

ABSTRACT

Although computational power is increasingly available, high-fidelity simulation based aerodynamic shape optimization is still challenging for industrial applications. To make the simulation based optimization acceptable in the practice of engineering design, a technique combining mesh morphing and reduced order modeling is proposed for efficient aerodynamic optimization based on CFD simulations. The former technique avoids the time-consuming procedure of geometry discretization. And the latter speeds up the procedure of field solution by combining pre-computed solution snapshots. To test the efficiency of the proposed method, the windshield of a motorbike is analyzed and optimized. It is shown that even the total number of cells of the mesh is around 0.4 million, the CFD computation and the post processing of the results can be completed in less than 10 seconds if the reduced order model is adopted. Running on a personal computer, the generic algorithm is applied to optimize the profile of the windshield. A 8% reduction of the drag coefficient is achieved after 800 queries of the reduced order CFD model and the total CPU time is only around 2 hours.

1. INTRODUCTION

High-fidelity modeling and simulation has been widely used in nowadays engineering design. In vehicle engineering, for example, detailed computational fluid dynamics (CFD) simulation is routinely applied to evaluate the aerodynamic performance of the vehicles. However, simulation based shape optimization is still a challenging task even the computational power is increasingly available because more complex and accurate models quickly use up the computational resources.

High-fidelity modeling techniques such as the finite element method and the finite volume method adopts finely resolved meshes which inevitably form big-sized algebra equations. Preparation of the detailed meshes as well as formation and solution of the algebra equations are the most time consuming steps in a simulation. For a non-trial industrial case, preparation of the mesh may take several or tens of hours even automatic mesh generation techniques are applied. Formation and solution of the algebra equations may take O(10-102) CPU-hours. As

*Corresponding Author: zhangchy5@mail.sysu.edu.cn

a consequence, the total computation cost of the simulation-based optimization is prohibitively high considering O(102-103) simulations have to be run in the optimization procedure.

To make the simulation-based optimization acceptable in the practice of engineering design, new methodologies should be developed to overcome the bottleneck of mesh generation and field solution. It is interesting to note that in typical shape optimization problems, the geometric configurations often change smoothly and can be adequately described by geometry morphing which in fact is driven by parameters. By applying the same morphing technique, the meshes can be directly generated from a mesh template (Fig.1), skipping out the time-consuming and error-prone process of geometry discretization. Efficient methodologies can also be developed to speed up the procedure of field solution (more specifically, formation and solution of the algebra equations) by combining pre-computed solution snapshots defined on well-chosen geometric configurations, thus avoiding the heavy computation of the algebra equations and enabling to perform simulations of complex phenomena almost in real time. This is the basic idea of reduced order modeling (ROM) [1]. Although construction of the pre-computed solutions is time-consuming, this step needs to be performed only once. The reduced order model built on top of the high-fidelity solution is fast-running and repeated calls to the ROM within the optimization procedure is acceptable.

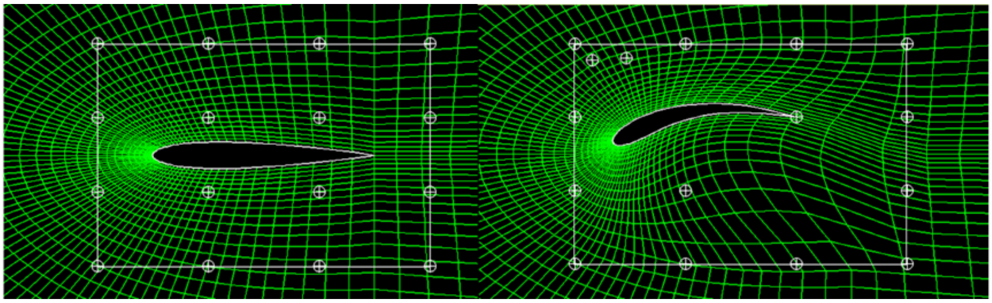

Figure 1 Schematic of mesh morphing of a two-dimension airfoil. The left is the original mesh and the right is the morphed mesh. The white hollow circles denote the control lattice.

In the present paper, we propose to combine the two techniques, i.e., the mesh morphing and the reduced order modeling, to formulate an automatic procedure for efficient aerodynamic shape optimization based on CFD simulations. The mesh morphing algorithm and a non-intrusive method for constructing reduced order CFD models is explained in Section 3. This nonintrusive choice allows us to apply the method to different shape optimization problems, changing only the high fidelity solver or the parametrization technique. In Section 3, implementation of the two technical ingredients is presented. Then the optimization of a motorbike by using the present method is illustrated in Section 4.

2. FREE FORM MESH MORPHING

Free form deformation (FFD) is a morphing technique widely used both in academia and in industry. The basic idea of FFD is to embed the part of the mesh (or the geometry) to be morphed in a control lattice and to deform it using a trivariate tensor-product of Bézier or B-spline functions[2]. By controlling the lattice points, a continuous and smooth deformation can be achieved (Figure 1). A typical FFD procedure usually consists of three steps. First, the physical domain Ω is mapped to the reference domain $\widehat{\Omega}$ through the map ψ. Then, some control points P of the lattice are moved to achieve the desired deformation using the map $\widehat{T}$. The displacements of such points which drive the deformation can be represented by a parameter vector μ. Finally the back mapping from the deformed reference domain is applied to deform the physical domain $\Omega(\boldsymbol{\mu})$ by the map ψ^{-1}.

A FFD-based morphing actually changes only the coordinates of the vertices located in the control box. It is independent from the topology of the object and extremely versatile and suitable to parametrize very complex geometries, including volume meshes, surface triangulations and CAD representations. In the present study, it is applied directly to deform the volumetric mesh by changing the positions of the influenced vertices. The elemental connectivity and the other properties of the mesh are left untouched. To avoid the deformed elements overlap the undeformed ones, it is suggested to move only the internal control points of the lattice. In the case of shape optimization, FFD on mesh allows to skip the time-consuming procedure of geometry discretization for every new geometric configuration, contributing to a significant time saving.

3. REDUCED ORDER CFD MODELING

The present study focuses on the aerodynamic performance of vehicles and the full order model is represented by the resolution of the well-known Navier-Stokes (NS) equations, i.e., a system of coupling differential equations which govern the dynamics of fluids. The Navier-Stokes equations are the most accurate continuum-based approximation for viscous flows where both convective and diffusive effects contribute, and they are known to accurately reproduce many interesting physical phenomena observed in fluids, such as the onset of turbulence. In vehicle engineering, the velocity of vehicles is high and turbulent flow often occurs which is characterized by significant and irregular variations in space and time. This brings further difficulties to high-fidelity CFD modeling. Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) are very accurate for turbulence modeling. However, they are computationally expensive and as an alternative, the Reynolds-averaged Navier–Stokes (RANS) equations are sufficient to describe the main time-averaged properties of the flow for most engineering applications. The general idea behind the RANS approach is to decompose velocity U and pressure p into ensemble-averaged and fluctuating components (Reynolds decomposition), obtaining approximate solutions to the NS equations. A turbulence model (e.g., Spalart–Allmaras, k-ε, k-ω)[3] is required to provide closure to the system of equations and to compute the Reynolds stress, the unknown term which accounts for fluctuations contribution.

The full order model can be solved by several standard discretization techniques such as the finite element (FE) method, the finite volume (FV) method and the finite difference (FD) method. Even if the FE method can be more accurate, the FV method is most widely chosen for industrial applications since it is easily applicable to realistic and physical context. Nowadays, the most majority CFD codes used in the industry, both commercial (CFX, FLUENT, STARCD) [4] (Iaccarino, 2001) and open-sourced (OpenFOAM) [5], are based on FV discretization. Non-trivial industrial applications may contain millions of even billions of computational cells, making high-fidelity CFD simulation unaffordable to many-query or real time control applications. Reduced Order Models (ROMs) have been proposed as a way of overcoming the computational burden required to obtain high fidelity solutions in large-scale systems.

3.1. Proper orthogonal decomposition

The main assumption of reduced order modeling is that the behavior of the system with respect to the physical or the geometric parameters can be represented by a small number of dominant modes. Among the several reduced order techniques, the Proper Orthogonal Decomposition (POD) with the snapshot technique[6,7] is probably the most widespread in the complex fluid flow computation. The POD technique was first introduced to study the coherent structures in experimental turbulent flows [8-10] but it has recently become a valuable option in the ROM framework [11-14] due to the capability to select the most energetic modes representing the most significant features of the system.

POD in the context of CFD modeling usually is based on snapshots which are state solutions computed at different instants in time and/or different parameter values. Consider a set of ns snapshots, u1, u2,…, uns. Here $\boldsymbol{u}_j = \boldsymbol{u}(t_j; \boldsymbol{\mu}_j) \in \mathcal{R}^n$ denotes the jth snapshot, where tj and μj are respectively the time and the parameter values for the jth snapshot. Define the snapshot matrix $\mathbf{U} \in \mathcal{R}^{n \times n_s}$, which contains the snapshot uj as its jth column. The left singular value decomposition of U is written as,

$$U = X \Sigma \mathrm{Y}^T \tag{1}$$

where the columns of the matrices $\mathbf{X} \in \mathcal{R}^{n \times n_s}$ and $\mathbf{Y} \in \mathcal{R}^{n_s \times n_s}$ are the left and right singular vectors of U, respectively. $\boldsymbol{\Sigma} \in \mathcal{R}^{n_s \times n_s} = diag(\sigma_1, \sigma_2, \dots, \sigma_{ns})$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{ns}$, are the singular values of U, referred to as the POD singular values. The POD basis, $\boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_N]$ is then defined as the N left singular vectors of U that correspond to the N largest POD singular values. This yields an orthonormal basis. The POD provides an efficient low-dimensional representation of the snapshot data: among all the orthonormal bases of size N, the POD basis minimizes the least squares error of snapshot reconstruction.

The square of the error in snapshot representation is given by the sum of the squares of the singular values corresponding to those singular vectors not included in the POD basis. Thus, the singular values provide a quantitative guidance for choosing the size of the POD basis. Since the POD basis is constructed from the sampled solutions, the POD method makes no assumptions about the form of the full model, i.e., POD applies to both linear and nonlinear systems, as well as to parametrically-varying systems.

3.2. POD with interpolation

Exploiting the new basis, the full order solution can be expressed as a linear combination of the POD basis,

$$\boldsymbol{u}_n = \textstyle\sum_{i=0}^{N} \alpha_i \psi_i \tag{2}$$

For each new value of the parameter, the solution can be quickly sought if the value of the coefficients αi can be determined by some efficient techniques, for example, by interpolation.

Proper Orthogonal Decomposition with Interpolation (PODI) was first introduced by Bui-Thann [15] and has been used recently in aerodynamic application [16]. The basic idea behind PODI is very simple, i.e., evaluation of the new coefficients by interpolation of the already computed coefficients for the parameter samples $\boldsymbol{\mu}_k \in \Xi$. For these parameter samples, the full order solutions have been obtained and through POD, the POD coefficients for each of the parameter can be constructed by projecting the full order solution onto the low order POD space,

$$\forall \boldsymbol{\mu}_k \in \Xi: \boldsymbol{u}(\boldsymbol{\mu}_k) = \boldsymbol{u}^N(\boldsymbol{\mu}_k) = \textstyle\sum_{i=0}^{N} \alpha_i(\boldsymbol{\mu}_k)\psi_i \tag{3}$$

For each new value of the parameter $\boldsymbol{\mu}_{new}$, the new coefficients $\alpha_i(\boldsymbol{\mu}_{new})$ can be obtained by interpolating the coefficients of the parameter samples, i,e., $\alpha_i(\boldsymbol{\mu}_k)$. Once the new coefficients are knowns, the reduced order solution can be readily constructed by using the POD basis,

$$\boldsymbol{u}_{new}^N = \textstyle\sum_{i=0}^{N} \alpha_i(\boldsymbol{\mu}_{new})\psi_i \tag{4}$$

An efficient and robust technique for N-th dimensional interpolation is based on the radial basis function (RBF) [17]. The RBF interpolation is well suitable for data points irregularly distributed in space (Fig. 2). In its basic form, RBF interpolation is in the form,

$$\mathrm{y}(\vec{x}) = \textstyle\sum w_i \varphi(|\vec{x} - \vec{x}_i|) \tag{5}$$

where the approximating function $\mathrm{y}(\vec{x})$ is represented as a sum of N radial basis functions φ, each associated with a different center $\vec{x}_i$, and weighted by an appropriate coefficient wi. denotes the distance between any point $\vec{x}$ and the center $\vec{x}_i$. For distance, one usually chooses the Euclidean distance. The weights wi can be estimated using the matrix methods of linear least squares, because the approximating function is linear in the weights. Note that the centers $\vec{x}_i$ can be located at arbitrary points in the domain, and do not require a grid. There are many forms of radial basis functions and in the present study, the Gaussian form is chosen,

$$\varphi(r) = exp\left(-\frac{1}{2}\frac{r^2}{r_0^2}\right) \tag{6}$$

where r_0 controls the size of the influence domain.

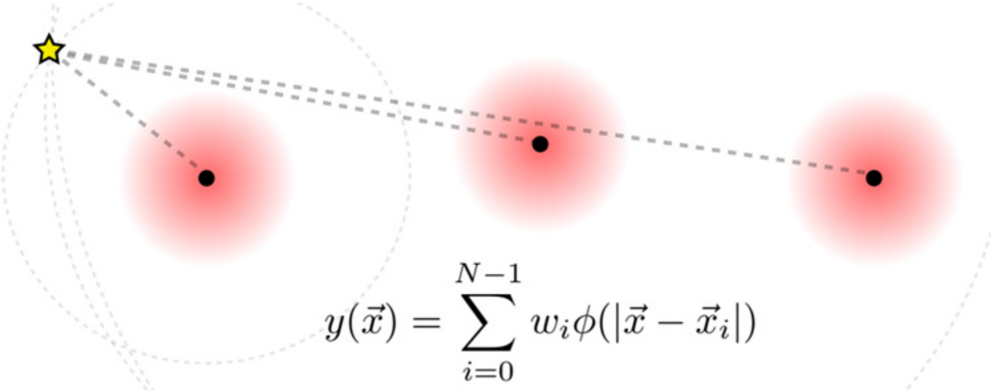
$y(\vec{x}) = \sum_{i=0}^{N-1} w_i \phi(|\vec{x} - \vec{x}_i|)$

Figure 2 Schematic of RBF interpolation

To limit the error induced by the low-order projection, the POD basis can be enriched by adding more full order solution snapshots to the snapshot matrix. The new parameter to generate the enriched solution is estimated by interpolating the existing parameter samples with weights indicating the approximation quality of each snapshot. The weights are evaluated by computing the error between the full order solution and the reconstructed reduced order solution of which the POD basis is generated with the snapshot matrix excluding the full order solution snapshot [18].

4. IMPLEMENTATION

In the present study, the open-sourced CFD solver, i.e., OpenFOAM, is chosen to generate the full order solution snapshots. OpenFOAM solves the NS equations using the finite volume discretization of which the coordinates of the vertices, the elemental connectivity as well as other information of the mesh are respectively defined by files. Considering the mesh morphing operates on vertices, only the file defining the vertices (i.e., the file named ‘points’) is parsed during the free form mesh morphing. A python script is developed to read the coordinates of the vertices and do the free form deformation morphing according to the user defined control lattice. The file defining the vertices is then overwritten with the new vertices.

The key step of the proposed method is to build the ROM parameterized with the external shape of the vehicle. With free form morphing, the parametric external shape is in fact defined in terms of the displacements of the control points within the lattice. To evaluate the drag force, the velocity, the pressure as well as the eddy viscosity should be constructed. Therefore, it is necessary to build the POD basis for the velocity (in fact, the components of the velocity), the pressure as well as the turbulence viscosity. In the present study, the two-equations k-ω model is chosen for the turbulence modeling. All the POD basis for each of the solution fields are computed on the same snapshot matrix which are enriched until a specified error tolerance is reached. Once the solution fields are constructed, the drag and the lift coefficients can be computed by calling the post-processing utility of OpenFOAM.

A python package, i.e., pyOpt, is used to minimize the objective function which accepts the displacements of the control points as inputs and returns the drag coefficient as output.

5. NUMERICAL TESTS

In the present study, the aerodynamic performance of a motorbike (Fig.3) is analyzed and optimized. The length, the width and the height of the motorbike are 2.0m, 0.6m and 1.35m, respectively. The box of the control lattice encloses the windshield which has a noticeable influence on the aerodynamic performance. To avoid any overlapping of the elements in the deformed mesh, only the eight internal points (hollow dots as shown in Fig.3) are free to move.

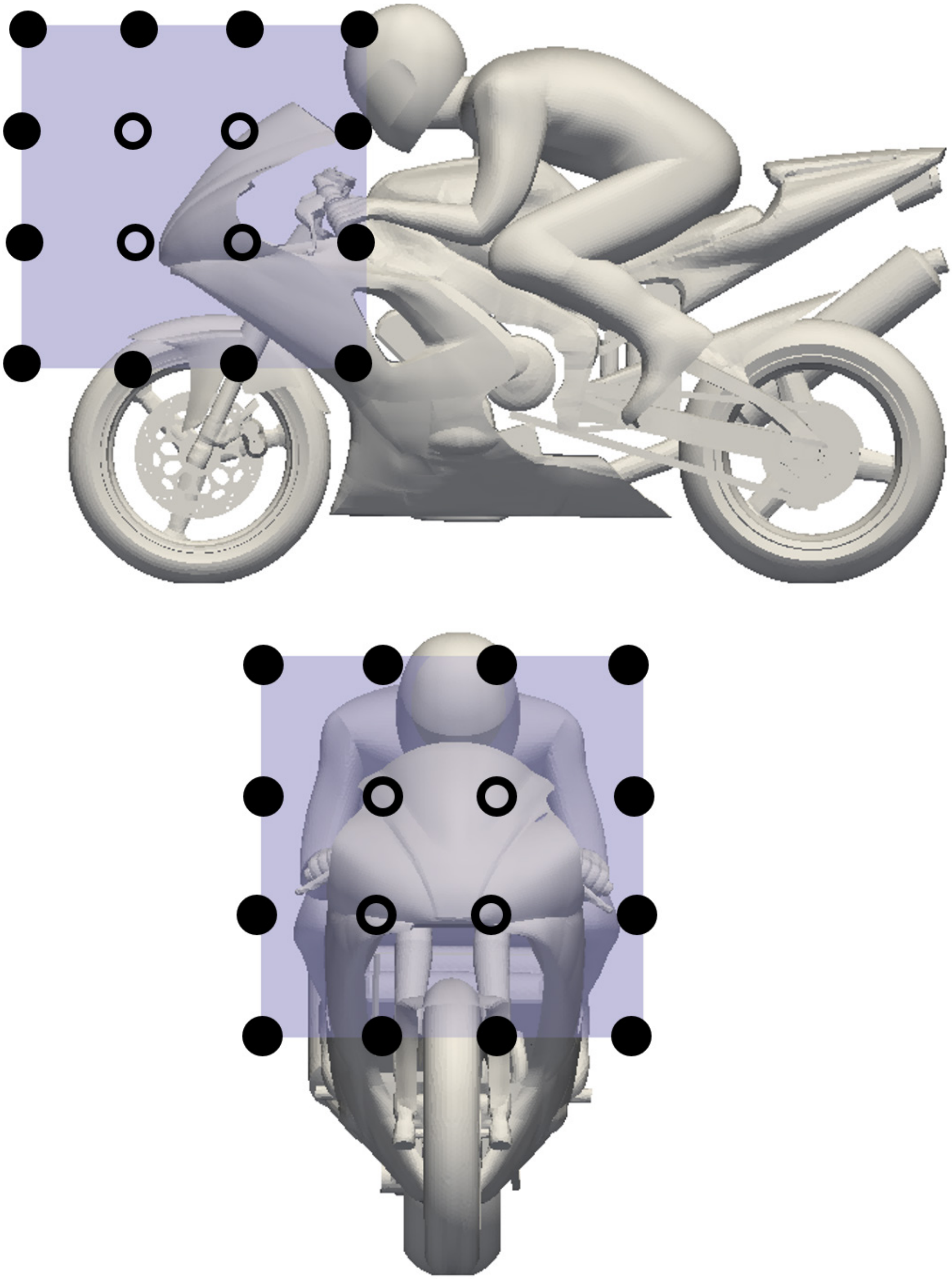

Figure 3 Schematic of the motorbike and the box of the control lattice.

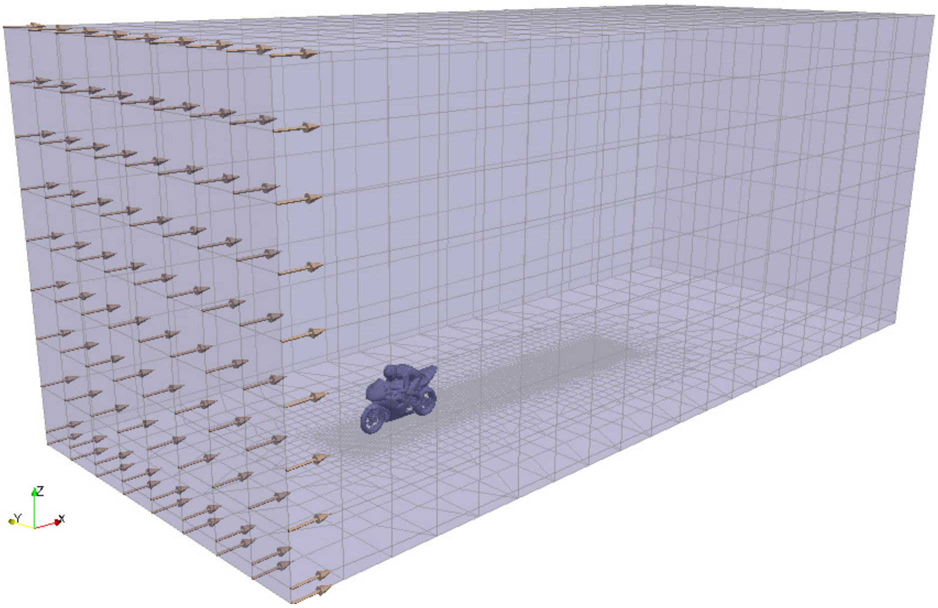

Figure 4 Illustration of the mesh of the computational domain.

The mesh of the whole computation domain is illustrated in Fig.4. The mesh is refined around the motorbike and a boundary layers is defined. Totally around 0.4 million points, 0.35 million cells and 1.1 million faces are generated. The inlet velocity is fixed at 20m/s.

The steady-state solver for incompressible, turbulent flow, i.e., the 'simpleFoam' solver, is chosen for the full order solutions. The time step is set to be 1s and the total steps are 500. The residual tolerance for all the solutions is set to be 10-8. It is checked that a steady state solution is ensured.

5.1. Construction of the ROM

Considering the symmetry of the model, the control points are also symmetrically positioned. Only one half of the internal points are needed to be considered. The width of the windshield (i.e., the size in the Y direction) is keep fixed and there are totally 8 parameters which control the shape of the windshield (Table 1). The Latin hypercube sampling method is used to generate the initial 16 parameter vectors.

Table 1 Parameters of the model

Index of the control point	**Scope of displacements (unit: m)**
(2,2,2)	ΔX1:(-0.1,0.1), ΔZ1:(-0.1,0.1)
(3,2,2)	ΔX2:(-0.1,0.1), ΔZ2:(-0.1,0.1)
(2,3,2)	ΔX3:(-0.1,0.1), ΔZ3:(-0.1,0.1)
(3,3,2)	ΔX4:(-0.1,0.1), ΔZ4:(-0.1,0.1)

For each of the parameter vector, the solutions at the final time step are collected. All the computations are performed in parallel on a HP Z400 workstation with 12 CPU cores. The error limit for enriching the sample is set to be 10-3 and finally additional 13 full order solutions are supplemented to the snapshot matrix. It takes around 4 hours to complete one full order solution and the total CPU time for constructing the ROM is 121 hours which include the solution of the 29 snapshots, the POD and the PODI.

5.2. Optimization by exploiting the ROM

Considering the fields of the velocity, the pressure, the turbulent kinetic energy and the turbulence frequency can be reconstructed in nearly real time, the genetic algorithm (GA) is chosen to do the optimization although it is not the most efficient method. The size of population of GA is set to be 20 and 40 generations of evolution are performed. It takes around 10s to complete one evaluation and totally it takes around 2 hours to complete the optimization procedure thanks to the fast-running ROM. The change of the drag coefficient during the evolution is shown in Fig.5. It is seen that 40 generations of evolution is enough to achieve the stabilization. An 8% reduction of the drag coefficient is achieved, and the optimized shape is shown in Fig.6.

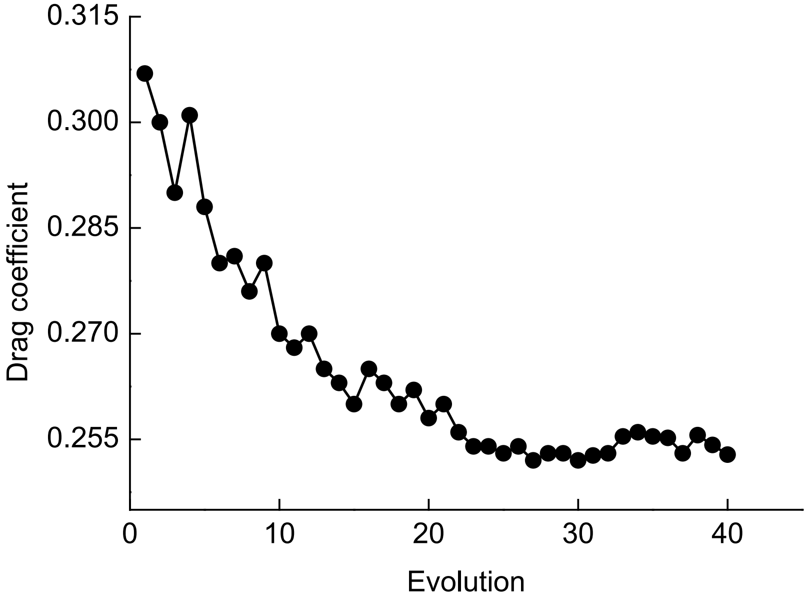

Figure 5 Decrease of the drag coefficient with the number of evolutions

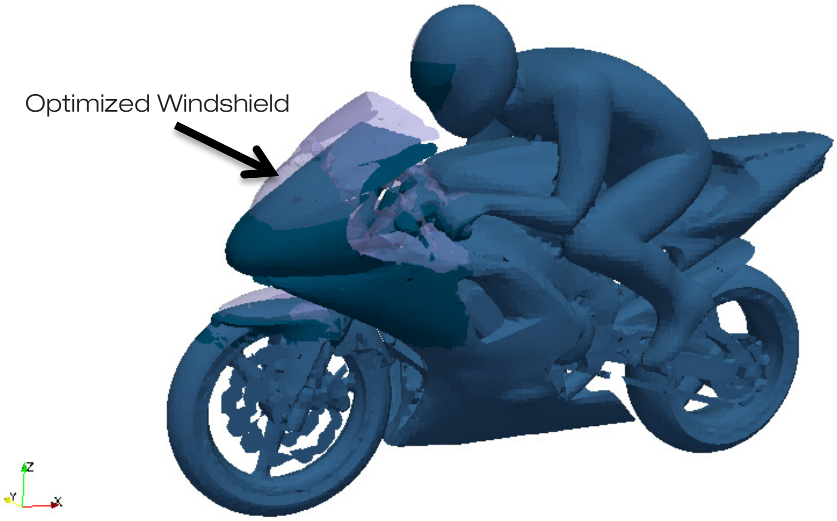

Figure 6 Optimized profile of the windshield compared with the original profile.

6. SUMMARY

The present proposes a chain of techniques for aerodynamic shape optimization by integrating free form mesh morphing, POD and PODI. The method is tested on a non-trivial industrial model. The key of the method is reconstruction of the solution fields with acceptable errors and a considerable speed-up by using the ROM. Another advantage is that the method does not rely on a particular discretization method. In fact, the PODI approach treats the high-fidelity solver completely as a black box. This feature allows to exploit the user preferred software, even commercial ones. Therefore, the method is also applicable to other industrial fields. One improvement to be made is to replace the conventional POD with the incremental POD in order to lower the memory usage of building ROM.

DISCLOSURE STATEMENT

No potential conflict of interest was reported by the authors.

ACKNOWLEDGEMENT

This work was supported by the Guangdong Provincial Science and Technology Project (Industry-University-Research Collaboration), grant number 2015B090901051.

REFERENCES

[1] Lassila T., A. Manzoni, A. Quarteroni and G. Rozza.Model Order Reduction in Fluid Dynamics: Challenges and Perspectives.Reduced Order Methods for Modeling and Computational Reduction, edited by Alfio Quarteroni and Gianluigi Rozza. 2014, pp.235-273. Springer.

[2] Sederberg T. W., and S. R. Parry.. Free-form Deformation of Solid Geometric Models. ACM SIGGRAPH computer graphics 20(1986),pp.151-160.

[3] Versteeg H.K., W. Malalasekera, An Introduction to Computational Fluid Dynamics, the Finite Volume Method, Pearson Education Limited, 2007.

[4] Iaccarino G., Predictions of a Turbulent Separated Flow Using Commercial CFD Codes, J. Fluids Eng. 123(2001), pp.819–828.

[5] Weller H.G., G. Tabor, H. Jasak, C. Fureby,. A Tensorial Approach to Computational Continuum Mechanics using Object-oriented Techniques, Comput. Phys. 12(1998), pp.620–631.

[6] Holmes P., J.L. Lumley, G. Berkooz,. Turbulence, Coherent Structures, Dynamical Systems and Symmetry, 1996, Cambridge.

[7] Sirovich L.,. Turbulence and the Dynamics of Coherent Structures. Parts I–III, Quart. Appl. Math. 45(1987), pp.561–590.

[8] Aubry N., P. Holmes, J. Lumley, E. Stone, The Dynamics of Coherent Structures in the Wall Region of a Turbulent Boundary Layer, J. Fluid Mech. 192(1988), pp.115–173.

[9] Berkooz G., P. Holmes, J.L. Lumley,. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows, Annu. Rev. Fluid Mech. 25(1993), pp.539–575.

[10] Lumley J.L., 1967. The Structure of Inhomogeneous Turbulent Flows, in: Atmospheric Turbulence and Radio Wave Propagation 1967, pp. 166–178.

[11] Cazemier W., R.W.C.P. Verstappen, A.E.P. Veldman, Proper Orthogonal Decomposition and Low-dimensional Models for Driven Cavity Flows, Phys. Fluids 10(1998), pp.1685–1699.

[12] Kunisch K., S. Volkwein, Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fuid Dynamics, SIAM J. Numer. Anal. 40(2003), pp.492–515.

[13] Weller J., E. Lombardi, M. Bergmann, A. Iollo, Numerical Methods for Low-order Modeling of Fuid Flows Based on POD, Internat. J. Numer. Methods Fluids 63(2010),pp. 249–268.

[14] Wang Z., I. Akhtar, J. Borggaard, T. Iliescu, Proper Orthogonal Decomposition Closure Models for Turbulent Fows: a Numerical Comparison, Comput. Methods Appl. Mech. Engrg. 237–240(2012.), pp.10–26.

[15] Bui-Thanh, T., Proper Orthogonal Decomposition Extensions and Their Applications in Steady Aerodynamics, Ph.D. thesis. 2003. Singapore-MIT Alliance.

[16] Dolci, V., and R. Arina. Proper Orthogonal Decomposition as Surrogate Model for Aerodynamic Optimization", Int. J. Aerospace Eng. 3(2016), pp.1-15.

[17] Broomhead D.S. and D. Lowe. Multivariable Functional Interpolation and Adaptive Networks, Complex Systems 2(1988), pp. 321–355.

[18] Salmoiraghi F., A. Scardigli, H. Telib and G. Rozza,. Free-form Deformation, Mesh Morphing and Reduced-order Methods: Enablers for Efficient Aerodynamic Shape Optimization, Int. J. Comput. Fluid D. 32(2018), pp.233-247.