Closed loop geometry based optimization by integrating subdivision, reanalysis and metaheuristic searching techniques

Guanxin Huang^{1,2}, Hu Wang^{1,2*}, Guoping Wang^{3,4}, Guangyao Li^{1,2}

¹ State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body,
Hunan University, China

² Joint Center for Intelligent New Energy Vehicle, China

³ Graphics & Interaction Laboratory, EECS Department, Peking University, China

⁴ Beijing Engineering Technology Center of Virtual Simulation and Visualization,
China

Abstract A closed loop geometry based optimization method is developed in this work by integrating subdivision, reanalysis and metaheuristic searching techniques. For that purpose, subdivision surfaces using triangle meshes are simultaneously applied to CAD (computer aided design) modeling and CAE (computer aided engineering) analysis. In this framework, feature objects are defined by the subdivision model, so that the subdivision models can be controlled by geometric parameters. Global optimization with metaheuristic algorithms is then performed, and an independent coefficients (IC) reanalysis method is introduced to enhance the efficiency of optimization. In order to improve the accuracy of triangular meshes, the ES-FEM (edge-based smoothed finite element method) technique is employed for analysis work. The proposed approach is successfully tested in three real engineering geometry optimization problems.

Keywords: Geometry based optimization; Subdivision; Reanalysis; Independent coefficient technique; Metaheuristic algorithms

1 Introduction

In conventional structural design, two kinds of models are used: CAD (computer

* Corresponding author: Hu Wang, Mailbox: wanghu@hnu.edu.cn, Tel: 86-731-88821417, Fax: 86-731-88822051

aided design) and CAE (computer aided engineering) models. The CAE model is derived from the CAD model by meshing. Sequentially, when the simulation results based on the CAD model are obtained, designers should adjust the CAD model according to the analysis results. Finally, the modified CAD model should be re-meshed. Obviously, it is impossible to develop a closed loop geometry optimization system based on such framework. In this study, to overcome this defect, a subdivision based CAD/CAE optimization method is developed. Due to this characteristic, a closed loop optimization can be performed. In other words, geometric parameters can be optimized automatically without artificial interference. To further improve the efficiency of optimization, reanalysis methods are introduced. Thus, objective function and corresponding constraints can be evaluated efficiently.

The first important issue is to integrate the CAD and CAE models seamlessly. B-rep (Boundary representation) [1] is a widely used model representation in computer geometry design. Under the framework of B-rep, there are two different popular methods for geometric modeling: Non-Uniform Rational B-Spline (NURBS) and subdivision. For some historical reasons [2], NURBS are popular for CAD, and subdivision for animation and entertainment. Hughes proposed a CAD/CAE integration method based on NURBS, which is known as IsoGeometric Analysis (IGA) [3], and it has been well developed [4]. In the IGA, the shape function is chosen as the basis functions of NURBS. It not only integrates CAD and CAE seamlessly, but also creates a feasible way for structural optimization, because NURBS control points can be used as design variables. However, IGA based on NURBS is not perfect due to the quality of NURBS [5]. For example, (1) because basis functions do not satisfy the orthogonality condition, mistakes and overlap may exist in IGA meshes; (2) in some complex models, surfaces may be discontinuous among different patches. Compared with NURBS, subdivision has its own characteristics and advantages. NURBS are usually generated on the grid with a well-defined topology. Each node of the grid is a control point of NURBS. The continuity of these patches needs to be carefully handled by designers, especially at the intersection points of several patches.

Conversely, subdivision can be generated on any polygon meshes without topological constraints. Therefore, subdivision can represent freeform surfaces without stitching patches. As subdivision level increases, mesh converges to a specific spline surface except for some extraordinary vertices. A detailed description of subdivision can be found in Ref [6]. Recently, some scholars paid attention to the field of NURBS-compatible subdivision surfaces [2]. Many operations on NURBS have their counterpart on subdivision, which ensures that the subdivision technology is suitable for CAD applications [7]. For example, trimming is an important and difficult operation in the NURBS based CAD system, and a bunch of methods have been developed for NURBS – symbolic, numerical or hybrid [8-10]. Trimming for subdivision surfaces is much simpler than for NURBS. The only concern is to constraint the boundary of the exact trim curve [11]. There are also methods for converting the trimmed NURBS surface to the Catmull-Clark subdivision surfaces [12].

More importantly, subdivision surfaces are usually based on simple polygons such as triangle (Loop subdivision [13], $\sqrt{3}$ subdivision [14]) and quadrangle (Doo-Sabin subdivision [15], Catmull-Clark subdivision [16]). Therefore, the CAE model can be discretized by the subdivision easily. Such idea of integrating CAD/CAE by using subdivision has also been developed recently [17, 18]. However, triangle mesh (the simplest polygon) is not suitable for structural analysis. Hence, several methods were developed in order to improve the accuracy of triangle mesh, such as assumed natural strain (ANS) [19], discrete shear gap (DSG) [20] and edge-based smoothed finite element method (ES-FEM) [21]. Among them, the ES-FEM technique is used in this study to make triangle mesh more feasible.

Another important issue is to find the global optimum efficiently. In engineering optimization, objective functions and constraints are usually evaluated by large-scale simulations, such as FE (finite element) and CFD (computational fluid dynamics) analysis. With the increase of complexity of problems, computational cost of such a simulation based problem becomes more expensive. The popular strategy to

overcome this bottleneck is to use a surrogate model instead of the exact function evaluation. However, error on predicted response value cannot be estimated and is difficult to be controlled. Surrogate assisted optimization (SAO) may miss the real global optimum and find a local optimum or even wrong results. Therefore, reanalysis methods are employed to improve the efficiency of optimization in this work. Predicted response values are more accurate than surrogate models and much closer to exact function evaluations. For these reasons, reanalysis can be directly integrated with some evolutionary global optimization methods such as genetic algorithm (GA). Reanalysis methods can be mainly divided into two categories: direct methods (DMs) and approximate methods. The DMs are usually suitable for low-rank or local modifications. They can obtain an exact solution of the modified structures. As stated by Tuckerman [22], any DM that solves a system by modification of an original model is an explicit or implicit application of the Sherman-Morrison-Woodbury (SMW) formula [23-25]. Cheikh *et al.* presented a DM based on a Moore Penrose generalized inverse for static reanalysis of structures [26]. Huang *et al.* developed the independent coefficients (IC) method for large-scale problems [27]. Song *et al.* proposed a direct reanalysis algorithm by updating the triangular factorization in sparse matrix solution [28]. Compared with DMs, approximate methods usually do not obtain an exact solution. However, high-rank or global modifications can usually be well disposed by these algorithms. Combined approximation (CA) [29-31] might be the most popular approximate reanalysis approach. The CA method was originally developed for linear static problems [32], and then extended to multidisciplinary problems. Kirsch and Papalambros presented a reanalysis approach for geometric changes in structural systems based on the CA method [33]. Rong *et al.* extended Kirsch's method to a new effective modal reanalysis method for topological modification [34]. Kirsch *et al.* used the CA method to overcome the difficulty of repeating eigenproblem solutions for nonlinear dynamic reanalysis [35]. Chen *et al.* suggested a universal iterative combined approximation (ICA) approach for all types of topological modifications [36]. Gao *et al.* proposed an adaptive time-based global

reanalysis (ATGR) for dynamic problems based on CA method [37]. Zhang *et al.* applied the CA method to probabilistic analysis and reliability-based design optimization of large-scale structures [38]. Other approximate reanalysis methods were developed in recent years. Epsilon-algorithm was presented for both static problems [39] and dynamic problems [40]. Preconditioned conjugate gradient (PCG) method is a classical iterative algorithm. Kirsch *et al.* introduced this method for structural reanalysis [41], and Wu *et al.* developed PCG method for both removal of degrees of freedoms (DOFs) and added DOFs [42-44]. A matrix perturbation method was used by Yang *et al.* for structural modal reanalysis [45-46]. Wu *et al.* also developed a reanalysis method based on rational approximation for very large changes in design variables [47]. Zuo *et al.* suggested some alternative CA methods for static structural analysis [48-49]. Recently, Wang *et al.* developed a fast analysis platform called “Seen is Solution” for complex product design [50]. Reanalysis can be applied to structural optimization efficiently.

The major advantage of reanalysis methods is to predict the response of the modified structure instead of the complete analysis. However, the application of reanalysis methods in optimization is still not as popular as expected due to the gap between CAD and CAE models. The present reanalysis methods concern more about the non-geometric optimization problem. For example, Kirsch used the CA method to optimize cross-section of truss structures [51]. Huang *et al.* suggested to use the IC method to accelerate the process of topological optimization [27]. Cai *et al.* enhanced the efficiency of compliance optimization of a continuum with bimodulus material under multiple load cases [52].

In view of the above mentioned issues, the purpose of this study is to develop a closed loop geometry optimization procedure based on subdivision and reanalysis. The new method is tested in three benchmark problems. The rest of this paper is organized as follows. In Section 2, subdivision technology is briefly introduced and is applied in CAD/CAE integration. In Section 3, a reanalysis based geometry optimization approach is described. Section 4 describes the test problems and discusses

optimization results. Conclusions are summarized in Section 5.

2 Subdivision in integration of CAD/CAE

2.1 B-rep based subdivision

The basic idea of B-rep is to use boundaries to represent CAD models. Particularly, an entity can be represented by its surfaces, a surface by its edges, an edge by vertices, and a vertex can be defined by coordinates. The information of surfaces, edges and vertices are recorded according to their levels. The connections between different levels form a tree data structure. An illustration of B-rep model and its data structure is shown in Fig.1.

Fig.1 B-rep and its data structure

The edges and surfaces in a B-rep model can be represented using subdivision. Subdivision is a discrete modeling theory. The idea of subdivision is to refine the mesh in an iterative way, and to obtain smooth curves or surfaces at the limiting condition (Fig.2). Nowadays, many outstanding subdivision algorithms have been developed for different purposes. For example, Chaikin algorithm [53] is used to generate subdivision curves; Loop subdivision and Catmull-Clark subdivision [16]

can be used to represent surfaces. As being employed in this study, the Loop subdivision and $\sqrt{3}$ subdivision are introduced in this section.

Fig.2 Example of subdivision

Assume that the vertex of initial mesh is $\mathbf{p}_i^0, (i = 1, 2, \dots, n_0)$. After subdivision for k times, the vertex becomes $\mathbf{p}_i^k, (i = 1, 2, \dots, n_k)$, where n_k indicates the number of vertices on the rank- k subdivision mesh ($k=0$ indicates the initial mesh). On the rank- k subdivision mesh, assume that there are n vertices $\mathbf{q}_{i,j}^k, (j = 1, 2, \dots, n)$ connected to vertex $\mathbf{p}_i^k$.

Loop subdivision

The procedure of Loop subdivision [13] is briefly described as follows:

(1) Update the present vertices as the new vertices on the next rank of subdivision mesh:

$$\mathbf{p}_i^{k+1} = (1 - n\beta_n)\mathbf{p}_i^k + \beta_n \sum_{j=1}^n \mathbf{q}_{i,j}^k, (i = 1, 2, \dots, n_k), \quad (1)$$

where,

$$\beta_n = \frac{1}{64n} \left(40 - \left(3 + 2 \cos \left(\frac{2\pi}{n} \right) \right)^2 \right) \quad (2)$$

or

$$\beta_n = \frac{3}{n(n+2)}. \quad (3)$$

(2) Generate the new edge points (see Fig.3), the new edge points can be generated as

$$\mathbf{p}_i^{k+1} = \frac{3\mathbf{p}_{i,1}^k + 3\mathbf{p}_{i,2}^k + \mathbf{p}_{i,3}^k + \mathbf{p}_{i,4}^k}{8}, (i = n_k + 1, n_k + 2, \dots, n_{k+1}). \quad (4)$$

Fig.3 Generation of a new edge point

(3) Re-link the new points: the new vertices are linked with the corresponding new edge points, and the new edge points are linked with the surrounding new edge points. The process of Loop subdivision is illustrated in Fig.4.

Fig.4 Process of Loop subdivision

$\sqrt{3}$ subdivision

The procedure of $\sqrt{3}$ subdivision [14] is briefly described as follows:

- (1) Update the present vertices as the new vertices on the next rank of subdivision mesh:

$$\mathbf{p}_i^{k+1} = (1 - n\beta_n)\mathbf{p}_i^k + \beta_n \sum_{j=1}^n \mathbf{q}_{i,j}^k, (i = 1, 2, \dots, n_k), \quad (5)$$

where,

$$\beta_n = \frac{4 - 2\cos(2\pi/n)}{9n}. \quad (6)$$

- (2) Generate the new surface points as (see Fig.5)

$$\mathbf{p}_i^{k+1} = \frac{\mathbf{p}_{i,a}^k + \mathbf{p}_{i,b}^k + \mathbf{p}_{i,c}^k}{3}, (i = n_k + 1, n_k + 2, \dots, n_{k+1}). \quad (7)$$

Fig.5 Generation of a new surface point

(3) Re-link the new points: the new surface points are linked with the corresponding new vertices; the present edges are rotated to link the new surface points.

The process of $\sqrt{3}$ subdivision is illustrated in Fig.6.

Fig.6 Process of $\sqrt{3}$ subdivision

2.2 Definition of design variables

It is easy to note that the subdivision models using the triangular mesh can be directly used for FE analysis. However, the numbers of vertices, edges and surfaces in a real engineering model will be as large as hundreds of thousands or even larger. If all entities change independently, it is difficult to modify the model efficiently in practice. Therefore, vertices, edges and surfaces in subdivision models need to be packed as some more meaningful combinations, which are defined as feature objects as follows.

Key point

A model which is represented by the triangle mesh contains a set of hundreds of thousands of vertices. Generally, there would be a small subset S_v that depicts the important features including ridges, ravines, corners and boundaries. Vertices in the relative complement of the subset with respect to the total set are redundant information from the geometry perspective. Therefore, the vertices in subset S_v are defined as key points for their contribution to the key property in geometry. For example, as shown in Fig.7, the vertices V_1 , V_2 , V_3 and V_4 are key points, while V_5 , V_6 , V_7 and V_8 are not.

Feature line

As shown in Fig.7, not all of edges in a triangle mesh are important. Those edges involving important features compose feature lines. A feature line may be straight, circle or general discrete zig-zag curves on the mesh. It provides the important information of geometry properties locally or even globally. The intersection points of feature lines are key points. A feature line usually uses 2 key points as its boundary, and contains one or more edges jointed with each other. For example, the feature line l_1 in Fig.7 has 2 boundary key points V_1 and V_2 , and contains three edges E_1 , E_2 and E_3 .

Feature surface

A feature surface uses feature lines as its boundary, and it consists of a set of jointed surfaces. For example, as shown in Fig.7, the feature surface S has 4 boundary

feature lines l_1 , l_2 , l_3 and l_4 . This feature surface contains all triangle surfaces shown in Fig.7.

Fig.7 Feature objects

By constructing the mesh as such a hierarchical data model, the shape of model can be controlled from bottom to top. It means the key points can be used to drive feature lines they lie on, and then these feature lines can drive other vertices on surface to deform as user's desire [54].

Since the data structure of 'key points – feature lines – feature surfaces' has been formed in Section 2.1, there are series of parameters which could be used to control the shape of a structure. According to the characteristic of engineering optimization, the following features are chosen as most commonly used design variables:

(1) Coordinates of key points

Key points are the bottom feature object in the data structure. Theoretically, the whole model can be controlled by some key points. Technically, the coordinates of key points are used to control the shape of feature lines as shown in Fig.8.

Fig.8 Design variables – key points

(2) Parameters of curves

Feature lines consist of jointed edges, and are usually controlled by fitted curves. These curves can be straight line, circle, B-spline or any other parametric curves. The parameters of these parametric curves can be used to control the shape of feature lines. For example, the circle hole shown in Fig.9 is controlled by the radius of circle. Moreover, we can regard the center of circle as a key point so that the position of hole can also be optimized.

Fig.9 Design variables – parameters of curves

(3) Cross-section

Cross-section is a specific defined combined object because of its special status in model representation and structure design. When a structure is modeled by sweeping technique, the shape of cross-section and sweeping guideline can specify its key property. When a discrete triangle mesh has sweeping structure, cross-sections should be cut out before it can be defined as design variables.

As shown in Fig.10, an auxiliary plane is used to divide mesh and it is defined as

$$a_0x + a_1y + a_2z + D = 0, \quad (8)$$

where (a_0, a_1, a_2) is normalized plane normal. Two half spaces are obtained as $a_0x + a_1y + a_2z + D > 0$ and $a_0x + a_1y + a_2z + D < 0$, respectively. For a single triangle, there are three vertices $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$ and $P_3(x_3, y_3, z_3)$.

Substituting these vertices into the plane equation, if

$$\begin{aligned} & (a_0x_i + a_1y_i + a_2z_i + D) \cdot (a_0x_j + a_1y_j + a_2z_j + D) < 0, \\ & (i, j \in \{1, 2, 3\}, i \neq j) \end{aligned} \quad (9)$$

then it is known that there must be a point on the edge P_i, P_j lying on the auxiliary plane. A non-degenerate triangle has less than two intersection points on edges. If there are two intersection points on the same triangle, a short intersecting line can be obtained by linking these two intersection points, and the triangle will be splitted into two parts. After triangularization, the split areas will become the new triangle meshes. A whole intersection curve can be obtained by linking all the short intersection lines in series. However, the intersection curve may be comprised of many discrete segments for which it is difficult to handle shape. Therefore, some points with high curvature are detected as the key points to control the shape of the intersection curve by interpolating other points on the curve. Regarding the cross-section on mesh as a new feature line, the whole mesh can be deformed according to different strategies (Fig.11). An example of modifying the cross-section plane of a structure is shown in Fig 12.

Fig.10 Creation of cross-section plane

Fig.11 Mesh deformation strategies

Fig.12 Cross-profile deformation strategies

3 Reanalysis assisted optimization

Assume that $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]$ is the vector of design variables and $x_i, i=1,2,\dots,n$ represent geometric parameters such as coordinates of a point, radius of a circle, length of a line and so on. The geometry optimization problem can be expressed as

$$\begin{aligned}
& \min \quad f(\mathbf{x}) \\
& s.t. \quad \begin{cases} g_i(\mathbf{x}) = 0, i = 1, 2, K, n_e \\ h_i(\mathbf{x}) \geq 0, i = 1, 2, K, n_i \\ \mathbf{x} \in \mathbf{D}(\mathbf{x}) \end{cases}
\end{aligned} \tag{10}$$

where $f(\mathbf{x})$ is the objective function; $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are equality constraints and inequality constraints, respectively, which put limits on structural design in terms of motion interference, stress distribution, stiffness/compliance, material properties, size/layout/topology of the structure, etc.; $\mathbf{D}(\mathbf{x})$ defines the bounds of design variables.

For most of engineering optimization problems, the objective function and constraints are usually implicit functions, and are commonly evaluated by expensive simulations. This may lead to a low efficiency or even make the optimization process difficult to converge. An alternative way to save computation cost is to use surrogate models instead of exact evaluations. However, when surrogate model is used, prediction errors cannot be avoided due to the fitting model, especially for complicated nonlinear problems. Compared with SAOs, reanalysis methods are obviously more reliable because the response of modified structure needs to be evaluated by the reduced equilibrium equation. In this study, a direct reanalysis method – independent coefficients (IC) [27] is employed to evaluate the response of modified structure. To guarantee the performance of reanalysis, ES-FEM [20] is used to improve the accuracy of triangle mesh.

3.1 Independent Coefficients reanalysis method

In structural geometry optimization, the shape of the structure changes frequently. Since performing exact structural analysis for each design modification may drastically reduce the computational efficiency of the optimization search, reanalysis methods are introduced in this section. Since the nodes of the structure may move in the process of geometry optimization, which will make some reanalysis methods like

CA [31] unstable, the IC method [27] is applied in this study. In the process of optimization, the first iterative step can be analyzed using complete analysis, and the remaining steps can be analyzed using IC. The theory of IC is recalled in the following.

Assume that the initial equilibrium equation is

$$\mathbf{K}_0 \mathbf{r}_0 = \mathbf{F}, \quad (11)$$

where $\mathbf{K}_0$ is the initial stiffness matrix, $\mathbf{r}_0$ is the initial solution and $\mathbf{F}$ is the load vector. After modification, the new equilibrium equation is

$$\mathbf{K} \mathbf{r} = \mathbf{F}. \quad (12)$$

Assume that the solution of Eq. (12) can be rewritten as

$$\mathbf{r} = \mathbf{r}_0 + \Delta \mathbf{r}. \quad (13)$$

Then Eq. (12) becomes

$$\mathbf{K}(\mathbf{r}_0 + \Delta \mathbf{r}) = \mathbf{F}. \quad (14)$$

After transformation, Eq. (12) can be written as

$$\mathbf{K} \Delta \mathbf{r} = \boldsymbol{\delta}, \quad (15)$$

where

$$\boldsymbol{\delta} = \mathbf{F} - \mathbf{K} \mathbf{r}_0. \quad (16)$$

Pre-select a small tolerance ε , if $|\delta(i)| > \varepsilon, i = 1, 2, \dots, n$, where n is the number of total degrees of freedom (DOFs), the i -th row of $\mathbf{K}$ should be examined. If $\mathbf{K}(i, j)$ is not 0, the j -th DOF should be selected to be reanalyzed, and the serial number is recorded in the vector $\mathbf{S}_d$.

Assume that

$$\Delta \mathbf{r} = y_1 \mathbf{v}_1 + y_2 \mathbf{v}_2 + \dots + y_s \mathbf{v}_s = \mathbf{R}_b \mathbf{y}, \quad (17)$$

where $\mathbf{v}_i, i = 1, 2, \dots, s$ are the basis vectors, $y_i, i = 1, 2, \dots, s$ are the corresponding coefficients and s is the number of basis vectors, which equals to the number of

selected DOFs.

The basis vectors $\mathbf{v}_i, i=1,2,\dots,s$ can be calculated as follows: if the i -th selected DOF is DOF j , then

$$\mathbf{v}_i = [0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0]^T, \quad (1 \text{ is the } j\text{-th element of } \mathbf{v}_i). \quad (18)$$

Substitute Eq. (17) into Eq.(15), and premultiply both sides of the equation with $\mathbf{R}_b^T$, Eq.(15) becomes

$$\mathbf{K}_r \mathbf{y} = \boldsymbol{\delta}_r, \quad (19)$$

where

$$\begin{aligned} \mathbf{K}_r &= \mathbf{R}_b^T \mathbf{K} \mathbf{R}_b \\ \boldsymbol{\delta}_r &= \mathbf{R}_b^T \boldsymbol{\delta} \end{aligned} \quad (20)$$

Solve Eq. (19), $\mathbf{y}$ is obtained. Then substituting $\mathbf{y}$ into Eq. (17), $\Delta \mathbf{r}$ can be achieved. Afterwards, $\mathbf{r}$ can be obtained from Eq. (13).

3.2 ES-FEM

In the ES-FEM, the displacement field is expressed as the linear interpolation of the 3 nodes' displacements:

$$\mathbf{u} = \{u, v, w, \theta_x, \theta_y\}^T = [\mathbf{N}_1 \quad \mathbf{N}_2 \quad \mathbf{N}_3] \begin{Bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{Bmatrix}, \quad (21)$$

where $\mathbf{d}_i$ is the generalized displacement of node i , the corresponding expression is

$$\mathbf{d}_i = \{u_i, v_i, w_i, \theta_{xi}, \theta_{yi}\}^T. \quad (22)$$

$\mathbf{N}_i$ is the diagonal matrix of the linear shape function of node i .

The membrane strain $\boldsymbol{\varepsilon}_m$ and the curvature $\boldsymbol{\varepsilon}_b$ can be written as

$$\boldsymbol{\varepsilon}_m = \hat{\mathbf{B}}_m \hat{\mathbf{d}} = [\hat{\mathbf{B}}_{m1} \quad \hat{\mathbf{B}}_{m2} \quad \hat{\mathbf{B}}_{m3}] \begin{Bmatrix} \hat{\mathbf{d}}_1 \\ \hat{\mathbf{d}}_2 \\ \hat{\mathbf{d}}_3 \end{Bmatrix}, \quad (23)$$

$$\hat{\boldsymbol{\varepsilon}}_b = \hat{\mathbf{B}}_b \hat{\mathbf{d}} = \begin{bmatrix} \hat{\mathbf{B}}_{b1} & \hat{\mathbf{B}}_{b2} & \hat{\mathbf{B}}_{b3} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}_1 \\ \hat{\mathbf{d}}_2 \\ \hat{\mathbf{d}}_3 \end{bmatrix}, \quad (24)$$

where

$$\hat{\mathbf{B}}_{mi} = \begin{bmatrix} N_{i,x} & 0 & 0 & 0 & 0 \\ 0 & N_{i,y} & 0 & 0 & 0 \\ N_{i,y} & N_{i,x} & 0 & 0 & 0 \end{bmatrix}, \quad (25)$$

$$\hat{\mathbf{B}}_{bi} = \begin{bmatrix} 0 & 0 & 0 & 0 & N_{i,x} \\ 0 & 0 & 0 & -N_{i,y} & 0 \\ 0 & 0 & 0 & -N_{i,x} & N_{i,y} \end{bmatrix}, \quad (26)$$

where $N_{i,x}$ and $N_{i,y}$ indicate the corresponding partial derivatives, and the superscript ‘ $\wedge$ ’ indicates element local coordinate system $(\hat{x}, \hat{y}, \hat{z})$.

For shear strain $\boldsymbol{\varepsilon}_s$, the DSG method [33] is used to eliminate the shear-locking.

According to the DSG method, the shear strain can be expressed as

$$\hat{\boldsymbol{\varepsilon}}_s = \hat{\mathbf{B}}_s \hat{\mathbf{d}} = \begin{bmatrix} \hat{\mathbf{B}}_{s1} & \hat{\mathbf{B}}_{s2} & \hat{\mathbf{B}}_{s3} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}}_1 \\ \hat{\mathbf{d}}_2 \\ \hat{\mathbf{d}}_3 \end{bmatrix}, \quad (27)$$

where

$$\begin{aligned} \hat{\mathbf{B}}_{s1} &= \frac{1}{2A_e} \begin{bmatrix} 0 & 0 & b-d & 0 & A_e \\ 0 & 0 & c-a & A_e & 0 \end{bmatrix} \\ \hat{\mathbf{B}}_{s2} &= \frac{1}{2A_e} \begin{bmatrix} 0 & 0 & d & -\frac{bd}{2} & \frac{ad}{2} \\ 0 & 0 & -c & \frac{bc}{2} & -\frac{ac}{2} \end{bmatrix}, \\ \hat{\mathbf{B}}_{s3} &= \frac{1}{2A_e} \begin{bmatrix} 0 & 0 & -b & \frac{bd}{2} & -\frac{bc}{2} \\ 0 & 0 & a & -\frac{ad}{2} & \frac{ac}{2} \end{bmatrix} \end{aligned} \quad (28)$$

where

$$\begin{aligned} a &= x_2 - x_1, b = y_2 - y_1 \\ c &= x_3 - x_1, d = y_3 - y_1 \end{aligned} \quad (29)$$

As is shown in Fig.13, the internal smoothing domain Ω_k contains two sub-domains Ω_{k1} and Ω_{k2} . In order to perform the strain smoothing, the strains of the different

elements sharing edge k should be transformed to the same coordinate system. An edge local coordinate system $(\bar{x}, \bar{y}, \bar{z})$ is constructed as shown in Fig.13, where $\bar{z}$ is the average normal direction of two elements sharing edge k . The strains of element local coordinate system then can be transformed into edge local coordinate system as

$$\begin{aligned}\bar{\boldsymbol{\varepsilon}}_m &= \mathbf{R}_{m1} \mathbf{R}_{m2} \hat{\boldsymbol{\varepsilon}}_m \\ \bar{\boldsymbol{\varepsilon}}_b &= \mathbf{R}_{b1} \mathbf{R}_{b2} \hat{\boldsymbol{\varepsilon}}_b, \\ \bar{\boldsymbol{\varepsilon}}_s &= \mathbf{R}_{s1} \mathbf{R}_{s2} \hat{\boldsymbol{\varepsilon}}_s\end{aligned}\quad (30)$$

where the strain transforming matrixes $\mathbf{R}_{m1}$, $\mathbf{R}_{m2}$, $\mathbf{R}_{b1}$, $\mathbf{R}_{b2}$, $\mathbf{R}_{s1}$, and $\mathbf{R}_{s2}$ are defined as

$$\mathbf{R}_{m1} = \mathbf{R}_{b1} = \begin{bmatrix} c_{\bar{x}\bar{x}}^2 & c_{\bar{x}\bar{y}}^2 & c_{\bar{x}\bar{z}}^2 & c_{\bar{x}\bar{x}}c_{\bar{x}\bar{y}} & c_{\bar{x}\bar{y}}c_{\bar{x}\bar{z}} & c_{\bar{x}\bar{x}}c_{\bar{x}\bar{z}} \\ c_{\bar{y}\bar{x}}^2 & c_{\bar{y}\bar{y}}^2 & c_{\bar{y}\bar{z}}^2 & c_{\bar{y}\bar{x}}c_{\bar{y}\bar{y}} & c_{\bar{y}\bar{y}}c_{\bar{y}\bar{z}} & c_{\bar{y}\bar{x}}c_{\bar{y}\bar{z}} \\ 2c_{\bar{x}\bar{x}}c_{\bar{y}\bar{x}} & 2c_{\bar{x}\bar{y}}c_{\bar{y}\bar{y}} & 2c_{\bar{x}\bar{z}}c_{\bar{y}\bar{z}} & c_{\bar{x}\bar{x}}c_{\bar{y}\bar{y}} + c_{\bar{y}\bar{x}}c_{\bar{x}\bar{y}} & c_{\bar{x}\bar{z}}c_{\bar{y}\bar{y}} + c_{\bar{y}\bar{z}}c_{\bar{x}\bar{y}} & c_{\bar{x}\bar{x}}c_{\bar{y}\bar{z}} + c_{\bar{y}\bar{x}}c_{\bar{x}\bar{z}} \end{bmatrix}, \quad (31)$$

$$\mathbf{R}_{s1} = \begin{bmatrix} 2c_{\bar{x}\bar{x}}c_{\bar{z}\bar{x}} & 2c_{\bar{x}\bar{y}}c_{\bar{z}\bar{y}} & 2c_{\bar{x}\bar{z}}c_{\bar{z}\bar{z}} & c_{\bar{x}\bar{x}}c_{\bar{z}\bar{y}} + c_{\bar{z}\bar{x}}c_{\bar{x}\bar{y}} & c_{\bar{x}\bar{z}}c_{\bar{z}\bar{y}} + c_{\bar{z}\bar{z}}c_{\bar{x}\bar{y}} & c_{\bar{x}\bar{x}}c_{\bar{z}\bar{z}} + c_{\bar{z}\bar{x}}c_{\bar{x}\bar{z}} \\ 2c_{\bar{y}\bar{x}}c_{\bar{z}\bar{x}} & 2c_{\bar{y}\bar{y}}c_{\bar{z}\bar{y}} & 2c_{\bar{y}\bar{z}}c_{\bar{z}\bar{z}} & c_{\bar{y}\bar{x}}c_{\bar{z}\bar{y}} + c_{\bar{z}\bar{x}}c_{\bar{y}\bar{y}} & c_{\bar{y}\bar{z}}c_{\bar{z}\bar{y}} + c_{\bar{z}\bar{z}}c_{\bar{y}\bar{y}} & c_{\bar{y}\bar{x}}c_{\bar{z}\bar{z}} + c_{\bar{z}\bar{x}}c_{\bar{y}\bar{z}} \end{bmatrix}, \quad (32)$$

$$\mathbf{R}_{m2} = \mathbf{R}_{b2} = \begin{bmatrix} c_{\hat{x}\hat{x}}^2 & c_{\hat{y}\hat{x}}^2 & c_{\hat{x}\hat{x}}c_{\hat{y}\hat{x}} \\ c_{\hat{x}\hat{y}}^2 & c_{\hat{y}\hat{y}}^2 & c_{\hat{x}\hat{y}}c_{\hat{y}\hat{y}} \\ c_{\hat{x}\hat{z}}^2 & c_{\hat{y}\hat{z}}^2 & c_{\hat{x}\hat{z}}c_{\hat{y}\hat{z}} \\ 2c_{\hat{x}\hat{x}}c_{\hat{x}\hat{y}} & 2c_{\hat{y}\hat{x}}c_{\hat{y}\hat{y}} & c_{\hat{x}\hat{x}}c_{\hat{y}\hat{y}} + c_{\hat{x}\hat{y}}c_{\hat{y}\hat{x}} \\ 2c_{\hat{x}\hat{y}}c_{\hat{x}\hat{z}} & 2c_{\hat{y}\hat{y}}c_{\hat{y}\hat{z}} & c_{\hat{x}\hat{y}}c_{\hat{y}\hat{z}} + c_{\hat{x}\hat{z}}c_{\hat{y}\hat{y}} \\ 2c_{\hat{x}\hat{x}}c_{\hat{x}\hat{z}} & 2c_{\hat{y}\hat{x}}c_{\hat{y}\hat{z}} & c_{\hat{x}\hat{x}}c_{\hat{y}\hat{z}} + c_{\hat{x}\hat{z}}c_{\hat{y}\hat{x}} \end{bmatrix}, \quad (33)$$

$$\mathbf{R}_{s2} = \begin{bmatrix} c_{\hat{x}\hat{x}}c_{\hat{z}\hat{x}} & c_{\hat{y}\hat{x}}c_{\hat{z}\hat{x}} \\ c_{\hat{x}\hat{y}}c_{\hat{z}\hat{y}} & c_{\hat{y}\hat{y}}c_{\hat{z}\hat{y}} \\ c_{\hat{x}\hat{z}}c_{\hat{z}\hat{z}} & c_{\hat{y}\hat{z}}c_{\hat{z}\hat{z}} \\ c_{\hat{x}\hat{x}}c_{\hat{z}\hat{y}} + c_{\hat{x}\hat{y}}c_{\hat{z}\hat{x}} & c_{\hat{y}\hat{x}}c_{\hat{z}\hat{y}} + c_{\hat{y}\hat{y}}c_{\hat{z}\hat{x}} \\ c_{\hat{x}\hat{z}}c_{\hat{z}\hat{y}} + c_{\hat{x}\hat{y}}c_{\hat{z}\hat{z}} & c_{\hat{y}\hat{z}}c_{\hat{z}\hat{y}} + c_{\hat{y}\hat{y}}c_{\hat{z}\hat{z}} \\ c_{\hat{x}\hat{x}}c_{\hat{z}\hat{z}} + c_{\hat{x}\hat{z}}c_{\hat{z}\hat{x}} & c_{\hat{y}\hat{x}}c_{\hat{z}\hat{z}} + c_{\hat{y}\hat{z}}c_{\hat{z}\hat{x}} \end{bmatrix}, \quad (34)$$

where $c_{\bar{x}\bar{x}}$ is the cosine of the angle between $\bar{x}$ and x axes, and $c_{\hat{x}\hat{x}}$ is the cosine of the angle between $\hat{x}$ and x axes, etc.

Fig.13 Local coordinate systems of ES-FEM

Substitute Eqs. (23), (24) and (27) into Eq. (30), the strains in edge local coordinate system is

$$\begin{aligned}\bar{\boldsymbol{\varepsilon}}_m &= \mathbf{R}_{m1} \mathbf{R}_{m2} \hat{\mathbf{B}}_m \hat{\mathbf{d}} \\ \bar{\boldsymbol{\varepsilon}}_b &= \mathbf{R}_{b1} \mathbf{R}_{b2} \hat{\mathbf{B}}_b \hat{\mathbf{d}} \quad . \\ \bar{\boldsymbol{\varepsilon}}_s &= \mathbf{R}_{s1} \mathbf{R}_{s2} \hat{\mathbf{B}}_s \hat{\mathbf{d}}\end{aligned}\quad (35)$$

The transformation of displacements between the local coordinate system and the global coordinate system is

$$\hat{\mathbf{d}} = \mathbf{T} \mathbf{d}, \quad (36)$$

where $\mathbf{d}$ is the element displacement in global coordinate system, and $\mathbf{T}$ is the transforming matrix. Thus, the strains in edge local coordinate system can be expressed using displacement in global coordinate system as

$$\begin{aligned}\bar{\boldsymbol{\varepsilon}}_m &= \mathbf{R}_{m1} \mathbf{R}_{m2} \hat{\mathbf{B}}_m \mathbf{T} \mathbf{d} = \bar{\mathbf{B}}_m \mathbf{d} \\ \bar{\boldsymbol{\varepsilon}}_b &= \mathbf{R}_{b1} \mathbf{R}_{b2} \hat{\mathbf{B}}_b \mathbf{T} \mathbf{d} = \bar{\mathbf{B}}_b \mathbf{d} \quad . \\ \bar{\boldsymbol{\varepsilon}}_s &= \mathbf{R}_{s1} \mathbf{R}_{s2} \hat{\mathbf{B}}_s \mathbf{T} \mathbf{d} = \bar{\mathbf{B}}_s \mathbf{d}\end{aligned}\quad (37)$$

The smoothing operation can be simply given as a weighted average of strains. Therefore, the smoothed strains can be given as

$$\begin{aligned}
\bar{\boldsymbol{\varepsilon}}_m^k &= \frac{A_{k1}}{A_k} \bar{\mathbf{B}}_m^{k1} \mathbf{d}^{k1} + \frac{A_{k2}}{A_k} \bar{\mathbf{B}}_m^{k2} \mathbf{d}^{k2} = \bar{\mathbf{B}}_m^k \mathbf{d}^k \\
\bar{\boldsymbol{\varepsilon}}_b^k &= \frac{A_{k1}}{A_k} \bar{\mathbf{B}}_b^{k1} \mathbf{d}^{k1} + \frac{A_{k2}}{A_k} \bar{\mathbf{B}}_b^{k2} \mathbf{d}^{k2} = \bar{\mathbf{B}}_b^k \mathbf{d}^k, \\
\bar{\boldsymbol{\varepsilon}}_s^k &= \frac{A_{k1}}{A_k} \bar{\mathbf{B}}_s^{k1} \mathbf{d}^{k1} + \frac{A_{k2}}{A_k} \bar{\mathbf{B}}_s^{k2} \mathbf{d}^{k2} = \bar{\mathbf{B}}_s^k \mathbf{d}^k
\end{aligned} \tag{38}$$

where

$$\begin{aligned}
\bar{\mathbf{B}}_m^k &= \frac{1}{A_k} \sum_I A_{kl} \bar{\mathbf{B}}_m^{kl} \\
\bar{\mathbf{B}}_b^k &= \frac{1}{A_k} \sum_I A_{kl} \bar{\mathbf{B}}_b^{kl} \\
\bar{\mathbf{B}}_s^k &= \frac{1}{A_k} \sum_I A_{kl} \bar{\mathbf{B}}_s^{kl}
\end{aligned} \tag{39}$$

and A_k , A_{k1} and A_{k2} are the areas of the domains Ω_k , Ω_{k1} and Ω_{k2} , respectively. The summation indicates an assembly process in Eq. (39). The stiffness matrix of the smoothing domain Ω_k then can be achieved as

$$\mathbf{K}^e = \mathbf{K}_m^e + \mathbf{K}_b^e + \mathbf{K}_s^e, \tag{40}$$

where

$$\begin{aligned}
\mathbf{K}_m^e &= h \bar{\mathbf{B}}_m^{kT} \mathbf{D}_m \bar{\mathbf{B}}_m^k A_k \\
\mathbf{K}_b^e &= \bar{\mathbf{B}}_b^{kT} \mathbf{D}_b \bar{\mathbf{B}}_b^k A_k \\
\mathbf{K}_s^e &= h \bar{\mathbf{B}}_s^{kT} \mathbf{D}_s \bar{\mathbf{B}}_s^k A_k
\end{aligned} \tag{41}$$

and

$$\mathbf{D}_m = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \mathbf{D}_b = \frac{h^3}{12} \mathbf{D}_m, \mathbf{D}_s = \kappa \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix}. \tag{42}$$

In Eq. (42), E is the modulus of elasticity, ν is the Poisson's ratio, and h is the thickness of the shell elements, and

$$\kappa = \frac{\pi^2}{12} \text{ or } \frac{5}{6}, G = \frac{E}{2(1+\nu)}. \tag{43}$$

4 Test problems and discussion of optimization results

In order to verify the performance of the proposed approach, three test problems are solved in this study. The first and the second problems are presented to compare SAO and Reanalysis Assisted Optimization (RAO), therefore, two optimization strategies are considered:

(1) EGO-complete: a popular SAO – Efficient Global Optimization (EGO) method [55] is used as the main optimization process, and the objective function is evaluated by complete analysis. The Kriging model is employed as surrogate in this study, and the initial samples are generated using Latin hypercube DOE (Design of Experiments).

The correlation function between two arbitrary points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\theta r_{ij}^2), \quad (44)$$

where r_{ij} is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and the parameter θ is determined by

$$\theta = n_s \frac{1}{n_v}, \quad (45)$$

where n_s is the number of samples, and n_v is the number of variables.

(2) GA-reanalysis: a widely used evolutionary global optimization algorithm – GA – is used without surrogate, and the objective function is evaluated by using reanalysis. The GA toolbox of MATLAB is utilized, and the setting of options could be found in Table 5. The Number of Function Evaluations (NFEs) of GA-reanalysis strategy may be much larger than EGO-complete, since EGO requires only one function evaluation for each iteration while GA works on a population of designs. However, the efficiency of an evaluation could be improved by reanalysis method, and Single Evaluation Time (SET) may be much smaller than complete analysis. Therefore, Total Computational Cost (TCC) is expected to be acceptable.

The third example is utilized to show the performance of IC-based reanalysis in metaheuristic algorithm-based optimization. Therefore, this example is solved by using several metaheuristic algorithms, involving GA, Simulated Annealing (SA),

Particle Swarm Optimization (PSO) and Differential Evolution (DE). The IC reanalysis is engaged during the optimization process to improve the computational efficiency.

4.1 Longitudinal member of car frame

A half-body model of a car is shown in Fig. 14. Torsional rigidity is an important property of a car frame. Under this working condition, the longitudinal member acts as a cantilever beam. A simplified longitudinal member is shown in Fig.15. To simulate the working condition of cantilever beam, the nodes marked A in Fig.15 are fixed (some members of A are not shown in Fig.16), and a concentrated force is enforced on node B along the z -direction. The objective function to be minimized is the displacement of node C along the z -direction. The cross-section profile of the longitudinal member is shown in Fig.16. The shape of stiffening plate is chosen as the design variable. In particular, coordinates of key points a , b and c are defined as design variables that may vary in the domains Ω_a , Ω_b and Ω_c , respectively. The optimization problem is stated as

$$\begin{aligned} \min \quad & y = \left| u_z^C(a(x_a, y_a), b(x_b, y_b), c(x_c, y_c)) \right| \\ \text{s.t.} \quad & \begin{cases} a(x_a, y_a) \in \Omega_a \\ b(x_b, y_b) \in \Omega_b \\ c(x_c, y_c) \in \Omega_c \end{cases}, \end{aligned} \quad (46)$$

where u_z^C is the displacement of Node C along z -direction, and (x_a, y_a) , (x_b, y_b) and (x_c, y_c) are the coordinates of key points a , b and c , respectively.

Fig. 14 Half-body model of a car frame

Solutions solved by EGO-complete and GA-reanalysis methods are demonstrated in Fig. 16. The corresponding convergence curves are shown in Fig. 17. It can be found that the GA-reanalysis strategy achieved a better solution yielding a smaller value of u_z -displacement for node C. Table 1 indicates that the GA-reanalysis requires more function evaluations (i.e. 620 vs. 53) than EGO-complete but computational cost of each function evaluation is much cheaper for the present method. Consequently, the CPU time required by the GA-reanalysis drops to less than three times as that of EGO-complete. Furthermore, it can be seen from Fig. 17 that GA-reanalysis strategy converges within 10 iterations, and at the 6th iteration, a similar solution is obtained by using the EGO-complete method. In this sense, the computational cost of the GA-reanalysis is 1251.8s, and EGO costs 253.2s. Although the GA-reanalysis is more expensive than the EGO, it is still feasible for engineering application. Since design modifications covered the whole profile of element cross-section, the advantage of IC may not come out clearly from the presented results. Moreover, as GA may be parallelized, the computational cost of the GA-reanalysis optimization process could be further reduced.

In order to validate the proposed approach, the optimal design obtained by the GA-reanalysis is analyzed by using the complete analysis based on ES-FEM, IC

reanalysis and Kriging model. The corresponding values of cost functions are listed in Table 2. It can be found that the result obtained by the IC reanalysis is much closer to the complete analysis and more accurate than the EGO in term of the objective function. According to Fig. 18, it can be found that the excellent agreement between u_z -displacement fields calculated by the IC-based reanalysis and the complete analysis based on ES-FEM for the optimized design obtained by GA.

Fig.15 3D mesh of a longitudinal member of the car frame

Fig.16 Cross-section profile of car frame longitudinal member

Table 1 Computational cost of GA-reanalysis and EGO-complete methods for the car frame longitudinal element problem

Methods	initial samples	Optimization iterations	NFEs	SET (s)	TCC (s)
EGO-complete	12	41	53	13.89	736.17
GA-reanalysis	0	31	620	10.37	6432.40

Fig.17 Convergence curves recorded in the car frame longitudinal member optimization

Table 2 Comparison of accuracy of reanalysis and approximation methods for the car frame longitudinal element problem

Methods	Complete analysis (mm)	IC reanalysis (mm)	Kriging model (mm)
Objective function values	1.33E-04	1.34E-04	1.37E-04

Fig.18 Car frame longitudinal member problem: u_z -displacement field evaluated by complete analysis and IC reanalysis for the optimized design obtained by the GA-reanalysis approach

4.2 Car door frame

Stiffness of car door frame is an important issue. A car door model is usually generated by sweeping cross sections. Therefore, the stiffness of car door frame can be optimized by changing the shape of corresponding cross section. A car door model is shown in Fig. 19. The model is composed of 27,571 nodes and 53,261 elements, involving 165,426 DOFs. The stiffness of car door frame is objective function. The nodes marked A , B , C and D in Fig. 19 are fixed, and a concentrated force is enforced on node E along the $-y$ direction. The objective function to be minimized is the displacement (absolute value) of node E along y -direction. The location of the cross-section plane is also shown in Fig. 19, and the shape of cross-section plane is shown in Fig. 20. The shape of inner panel is the design variable. In particular, coordinates of key points a , b and c are defined as design variables that may vary in the domains Ω_a , Ω_b and Ω_c , respectively. The optimization problem is stated as

$$\begin{aligned}
\min \quad & y = \left| u_y^E(a(x_a, y_a), b(x_b, y_b), c(x_c, y_c)) \right| \\
s.t. \quad & \begin{cases} a(x_a, y_a) \in \Omega_a \\ b(x_b, y_b) \in \Omega_b \\ c(x_c, y_c) \in \Omega_c \end{cases}, \quad (47)
\end{aligned}$$

where, u_y^E is the displacement of node E along y -direction, and (x_a, y_a) , (x_b, y_b) and (x_c, y_c) are the coordinates of key points a , b and c , respectively.

This test problem was solved with the EGO-complete and GA-reanalysis methods and the optimized solutions are compared in Fig. 20. The corresponding convergence curves are shown in Fig. 21. It can be seen that the GA-reanalysis method achieves a better solution yielding a smaller value of u_y -displacement for node E while the EGO-complete method remains trapped in local minimum. However, the suggested method requires about twice the CPU time of EGO-complete to carry out the optimization process. Furthermore, it can be seen from Fig. 21 that a similar solution to the EGO-complete method has already been obtained by the GA-reanalysis strategy at 17th iteration. The computational cost of GA-reanalysis is about 2764.8s, rather close to the one of the EGO-reanalysis.

The optimal design obtained by GA-reanalysis was also analyzed by complete ES-FEM analysis, ES-FEM based IC reanalysis and Kriging model. The corresponding values of objective function are compared in Table 4. It can be found that the result obtained by IC reanalysis is almost same as the one by the complete analysis. The u_y -displacement field evaluated by FE analysis also is shown in Fig. 22 for the sake of comparison. It can be seen that FE results are quite different from the results of ES-FEM based analysis and reanalysis because of mesh distortion.

Fig.19 3D mesh of a car door

Fig.20 Cross-section profile of car door frame

Table 3 Computational cost of GA-reanalysis and EGO-complete methods for the car door frame problem

Methods	initial samples	Optimization iterations	NFEs	SET (s)	Total cost (s)
EGO-complete	12	41	53	42.08	2230.08
GA-reanalysis	0	31	620	8.64	5356.80

Fig.21 Convergence curves recorded in the car door frame optimization

Table 4 Comparison of accuracy of reanalysis/approximation methods for the car door frame problem

Methods	Complete analysis (<i>mm</i>)	IC reanalysis (<i>mm</i>)	Kriging model (<i>mm</i>)
Objective function values	2.22E-02	2.23E-02	2.27E-02

Complete analysis based on ES-FEM

Reanalysis based on ES-FEM

Complete analysis based on FEM

Fig.22 Car door frame problem: final optimized u_y -displacement field evaluated by ES-FEM based complete analysis, reanalysis and FE based analysis.

4.3 Design of Concave mold for U-shape parts

The third test problem solved in this study regards the concave mold model for U-shaped parts shown in Fig. 23. In order to reduce the weight of mold and preserve its stiffness and strength, size and position of main lightening holes must be optimized. Figure 24 shows the cross-section of one half of the mold model utilized in the optimization process. Boundary conditions are set as follows: (i) u_x -displacements equal to 0 on boundary B and u_y displacements equal to 0 on boundary C; (ii) uniform pressure of 1 N/m acting on boundary A. The optimization problem is stated as

$$\begin{aligned} \min \quad & y = \max(s(r, x_o, y_o)) \\ \text{s.t.} \quad & \begin{cases} c \leq 0.01 \\ \sigma_e \leq 27 \\ u_{\max} \leq 2.5 \times 10^{-5} \\ r \in [20, 30], x_o \in [16, 18], y_o \in [-29, -27] \end{cases}, \end{aligned} \quad (48)$$

where s is the area of the mold cross-section shown in Fig. 24; r , x_o and y_o are the radius and center coordinates of the main lightening circular hole O (see Fig. 24), respectively; c is the compliance of the structure defined as

$$c = \mathbf{u}^T \mathbf{K} \mathbf{u}, \quad (49)$$

where $\mathbf{u}$ is the displacement vector and $\mathbf{K}$ is the stiffness matrix of the structure. σ_e is the effective stress, and $u_{\max}$ is defined as

$$u_{\max} = \max(u). \quad (50)$$

Fig. 23 3D Model of concave mold for U-shaped parts

Fig. 24 Cross-section of the concave mold before and after GA-reanalysis optimization

Several metaheuristic algorithms (i.e. GA, SA, PSO and DE) are combined with IC reanalysis to solve this optimization problem. Convergence limit for objective function and constraint tolerance are set as 1.0×10^{-6} . The other internal parameters set for the different metaheuristic optimizers are listed in Table 5. Optimization results are presented in Table 6, which shows that similar solutions are obtained by the utilized approaches. Numerical efficiency is compared in Table 7. It appears that the IC-based reanalysis approach allowed to save much CPU time for metaheuristic search algorithms.

The optimal solution obtained by the GA-reanalysis, shown in Fig. 24, is then analyzed with complete analysis and IC reanalysis in order to verify the accuracy of the present approach. The corresponding deformation and effective stress fields are compared in Fig. 25. It appears that the results obtained by IC reanalysis and complete analysis are quite similar: the error on displacement is 2.59×10^{-4} . The error is evaluated as

$$\delta = \frac{\|\mathbf{r} - \mathbf{r}^*\|}{\|\mathbf{r}^*\|}, \quad (51)$$

where, $\mathbf{r}$ is the result obtained by IC reanalysis, and $\mathbf{r}^*$ is the result obtained by the complete analysis.

Table 5 Internal parameter setting for utilized metaheuristic searching algorithms

Algorithms	Parameters	
	Option	Value
GA	Population Size	50
	Crossover Fraction	0.8
	Others	Default in MATLAB
SA	Initial Temperature	100
	Temperature Function	100×0.95^k (k is the iteration number)
	Others	Default in MATLAB
PSO	Population Size	100
	Inertia Weight	1
	Inertia Weight Damping Ratio	0.99
	Personal Learning Coefficient	1.5
	Global Learning Coefficient	2
DE	Population Size	50
	Lower Bound of Scaling Factor	0.2
	Upper Bound of Scaling Factor	0.8
	Crossover Probability	0.2

Table 6 Comparison of optimization results for the concave mold problem

Algorithms	Solutions						
	Variables			Objective function value (mm^2)	Constraints value		
	r (mm)	x_o (mm)	y_o (mm)		Stress (kPa)	Compliance (μJ)	Displacement (mm)
GA	23.5723	16.0000	-28.7168	1.08E+04	-0.6404	-9.35E-07	-4.43E-06
SA	23.5324	16.0000	-28.3899	1.08E+04	-0.6363	-4.59E-14	-4.46E-06
PSO	23.5190	16.0837	-28.5848	1.08E+04	-0.6417	-1.20E-07	-4.43E-06
DE	23.5068	16.0000	-28.7816	1.08E+04	-0.6418	-1.58E-05	-4.46E-06

Table 7 Comparison of computational cost of reanalysis/approximation methods for
concave mold problem

Algorithms	CPU time (s) for one reanalysis	CPU time (s) for one complete analysis	NFEs	Saved CPU time (s)
GA			4757	3554.68
SA			2822	2108.39
PSO	6.02	13.49	1200	8961.00
DE			1400	10461.50

Displacement obtained by complete analysis

Displacement obtained by IC reanalysis

Stress obtained by complete analysis

Stress obtained by IC reanalysis

Fig. 25 Concave mold problem: u_y -displacement and effective stress field evaluated
with complete analysis and IC reanalysis for the optimized design obtained by the
GA-reanalysis approach

In order to examine the robustness of the proposed RAO frame work, GA is selected as an example to be tested. The GA based optimization of the concave mold is carried out for 100 times with random initial populations in uniform distribution, and the statistical data is shown in Table 8. From Table 8, it appears that GA based optimization is robust enough for engineering application.

Table 8 Statistical data of the GA based optimization of the concave mold

	$r \text{ (mm)}$	$x_o \text{ (mm)}$	$y_o \text{ (mm)}$	Objective function value (mm^2)
Mean value	23.59	16.00	-28.90	10810.87
Standard deviation	0.012	0.0003	0.11	1.79

4.4 Discussion

The optimization results of the first and the second test problems indicates that the GA-reanalysis method can find more accurate global optimum design than the EGO-complete method. However, the suggested method required much more function evaluations than the EGO-complete, because objective function must be evaluated repeatedly in each optimization iteration for the GA, but only once for the EGO. According to literature [26-27, 35], reanalysis may become effective when the optimization process entails just local modifications such as it happens in the car door frame problem: the speed-up ratio entailed by reanalysis for this case was about 6 and the total CPU time could be significantly reduced. Reanalysis should be effective also in case of large-scale problems allowing significant reductions in computational cost. Furthermore, parallel GA-reanalysis should be yet more feasible for geometry based design optimization than the serial version developed in this study.

Although the efficiency of IC reanalysis is not impressive compared with surrogate models, it is attractive while comparing with the complete analysis. From the third test problem, it can be seen that a significant fraction of computational cost has been saved by applying IC reanalysis. Considering the balance between efficiency and

accuracy, reanalysis may be a better choice than surrogate model for the global optimization method.

Moreover, since the IC is a reanalysis method for local modification, if the change of the geometric parameters will lead to global modifications, other suitable reanalysis methods should be considered.

5 Conclusions

In this study, an efficient closed loop geometry optimization system is developed by integrating subdivision and reanalysis methods as shown in Fig. 26. The contribution of this study to optimization literature is twofold:

Fig. 26 Framework of the proposed closed loop geometry optimization system

1. In order to realize automatic geometry optimization, a closed-loop process is developed by using a subdivision based CAD/CAE integration framework. The subdivision surfaces using triangular mesh are used for CAD modeling and CAE analysis simultaneously, therefore, CAD and CAE are associated automatically. Feature objects are defined on subdivision models, and by constructing a data structure of ‘key points – feature lines – feature surfaces’, the mesh model can be controlled by a set of geometric parameters.
2. The IC reanalysis method is integrated with metaheuristic algorithms in order to solve the geometry optimization problem at a rather low computational cost. The

feasibility of the proposed approach was clearly demonstrated by the results obtained in three geometry optimization problems. The efficiency of the RAO framework developed in this study increases if local regions of the design domain are interested by the optimization process. Furthermore, by parallelizing some metaheuristic algorithms such as GA, the present method may become very attractive also for large-scale optimization problems.

Acknowledgement

This work is supported by National Science Foundation of China (NSFC) under Grant Nos. 11172097, 61232014; National Key Technology Research and Development Program of China under Grant Nos. 2013BAK03B07 and 2015BAK01B06; Hunan Provincial Innovation Foundation for Postgraduate under grant Nos. 521293021 and CX2016B079.

Reference

1. F. R. Feito, J. C. Torres. Boundary representation of polyhedral heterogeneous solids in the context of a graphic object algebra. *The Visual Computer* 1997; 13:64-77.
2. T. J. Cashman. *NURBS-compatible subdivision surfaces*. University of Cambridge 2012.
3. T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; 194:4135-4195.
4. J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley and Sons 2009; 5:299-325.
5. V. P. Nguyen, C. Anitescu, S. P. A. Bordas, T. Rabczuk. Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation* 2015; 117: 89-116.
6. J. Peters, U. Reif. *Subdivision surfaces*. Springer Berlin Heidelberg, 2008.
7. W. Ma. Subdivision surfaces for CAD - An overview. *Computer-Aided Design* 2005; 37(7):693–709.
8. S. Krishnan and D. Manocha. An efficient surface intersection algorithm based on lower-dimensional formulation. *Acm Transactions on Graphics* 1997; 16(1):74–106.
9. J. Keyser, S. Krishnan, D. Manocha. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: I—representations.

- Computer Aided Geometric Design 1999; 16:861–882.
10. J. Keyser, S. Krishnan, D. Manocha. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: II — computation. Computer Aided Geometric Design 1999; 16:861–882.
 11. N. Litke, A. Levin, P. Schröder. Trimming for subdivision surfaces. Computer Aided Geometric Design 2001; 18(5):463–481.
 12. J. Shen, M. A. Sabin, N. A. Dodgson. Conversion of trimmed NURBS surfaces to Catmull – Clark subdivision surfaces. Computer Aided Geometric Design 2014; 31: 486–498.
 13. C. Loop. Smooth subdivision surfaces based on triangles. Master Thesis. University of Utah, 1987.
 14. L. Kobbelt. $\sqrt{3}$ -subdivision. Proceedings of Acm Siggraph 2000; 18(1):102-112.
 15. D. Doo, M. Sabin. Analysis of the behavior of recursive division surfaces near extraordinary points. Computer-Aided Design 1978; 10(6):356-360.
 16. E. Catmull, J. Clark. Recursively Generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 1978; 10(6):350-355.
 17. F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. Computer-Aided Design 2002; 34:137-148.
 18. C. Zhuang, J. Zhang, X. Qin, F. Zhou, G. Li. Integration of subdivision method into boundary element analysis. International Journal of Computational Methods 2012; 9(1):1240019 (11 pages).
 19. J. H. Kim, Y. H. Kim. A three-node C0 ANS element for geometrically non-linear structural analysis. Computer Methods in Applied Mechanics and Engineering 2002; 191:4035-4059.
 20. K. U. Bletzinger, M. Bischoff, E. Ramm. A unified approach for shear-locking-free triangular and rectangular shell finite elements. Computers and Structures 2000; 75:321-334.
 21. X. Y. Cui, G. R. Liu, G. Y. Li, G. Y. Zhang, G. Zheng. Analysis of plates and shells using an edge-based smoothed finite element method. Computational Mechanics 2010; 45: 141-156.
 22. L. S. Tuckerman. Divergence-free velocity fields in nonperiodic geometries. Journal of Computational Physics 1989; 80:403-441.
 23. J. Sherman, W. J. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. Annals of Mathematical Statistics 1949; 20(4):621.
 24. M. Woodbury. Inverting modified matrices. Memorandum Report, 42. Statistical Research Group. Princeton University, Princeton.
 25. M. A. Akgün, J. H. Garcelon, R. T. Haftka. Fast exact linear and non-linear structural reanalysis and the Sherman-Morrison-Woodbury formulas. International Journal for Numerical Methods in Engineering 2001; 50:1587-1606.
 26. M. Cheikh, A. Loredó. Static reanalysis of discrete elastic structures with

- reflexive inverse. *Applied Mathematical Modelling* 2002; 26:877-891.
27. G. Huang, H. Wang, G. Li. A reanalysis method for local modification and the application in large-scale problems. *Structural and Multidisciplinary Optimization* 2014; 49:915-930.
 28. Q. Song, P. Chen, S. Sun. An exact reanalysis algorithm for local non-topological high-rank structural modifications in finite element analysis. *Computers and Structures* 2014; 143:60-72.
 29. U. Kirsch. Combined approximations – a general reanalysis approach for structural optimization. *Structural and Multidisciplinary Optimization* 2000; 20:97-106.
 30. U. Kirsch, P. Y. Papalambros. Structural reanalysis for topological modifications – a unified approach. *Structural and Multidisciplinary Optimization* 2001; 21:333-344.
 31. U. Kirsch. A unified reanalysis approach for structural analysis, design, and optimization. *Structural and Multidisciplinary Optimization* 2003; 25:67-85.
 32. U. Kirsch, F. Moses. An improved reanalysis method for grillage-type structures. *Computers and Structures* 1998; 68:79-88.
 33. U. Kirsch, P. Y. Papalambros. Exact and accurate reanalysis of structures for geometrical changes. *Engineering with Computers* 2001; 17:363-372.
 34. F. Rong, S. Chen, Y. Chen. Structural modal reanalysis for topological modifications with extended Kirsch method. *Computer Methods in Applied Mechanics and Engineering* 2003; 192:697-707.
 35. U. Kirsch, M. Bogomolni, I. Sheinman. Nonlinear dynamic reanalysis of structures by combined approximations. *Computer Methods in Applied Mechanics and Engineering* 2006; 195:4420-4432.
 36. S. Chen, Z. Yang. A universal method for structural static reanalysis of topological modification. *International Journal for Numerical Methods in Engineering* 2004; 61:673-686.
 37. G. Gao, H. Wang, G. Li. An adaptive time-based global method for dynamic reanalysis. *Structural and Multidisciplinary Optimization* 2013; 48:355-365.
 38. G. Zhang, E. Nikolaidis, Z. P. Mourelatos. An efficient re-analysis methodology for probabilistic vibration of large-scale structures. *Journal of Mechanical Design* 2009; 131:1-13.
 39. Z. Yang, X. Chen, R. Kelly. An adaptive static reanalysis method for structural modifications using epsilon algorithm. *Computational Sciences and Optimization: Theory, Simulation and Experiment* 2009; 2:897-899.
 40. S. Chen, L. Ma, G. Meng. Dynamic response reanalysis for modified structures under arbitrary excitation using epsilon-algorithm. *Computers and Structures* 2008; 86:2095-2101.
 41. U. Kirsch, M. Kocvara, J. Zowe. Accurate reanalysis of structures by a preconditioned conjugate gradient method. *International Journal for Numerical Methods in Engineering* 2002; 55:233-251.
 42. B. Wu, C. Lim, Z. Li. A finite element algorithm for reanalysis of structures with added degrees of freedom. *Finite Elements in Analysis and Design* 2004;

- 40:1791-1801.
43. B. Wu, Z. Li. Reanalysis of structural modifications due to removal of degrees of freedom. *Acta Mechanica* 2005; 180:61-71.
 44. B. Wu, Z. Li. Static reanalysis of Structures with added degrees of freedom. *Communications in Numerical Methods in Engineering* 2006; 22:269-281.
 45. X. Yang, S. Chen, B. Wu. Eigenvalue reanalysis of structures using perturbations and Padé approximation. *Mechanical Systems and Signal Processing* 2001; 15(2):257-263.
 46. X. Yang, H. Lian, S. Chen. An adaptive iteration algorithm for structural modal reanalysis of topological modifications. *Communications in Numerical Methods in Engineering* 2002; 18:373-382.
 47. B. Wu, Z. Li, S. Li. The implementation of a vector-value rational approximate method in structural reanalysis problems. *Computer Methods in Applied Mechanics and Engineering* 2003; 192:1773-1784.
 48. W. Zuo, Z. Yu, S. Zhao, W. Zhang. A hybrid Fox and Kirsch's reduced basis method for structural static reanalysis. *Structural and Multidisciplinary Optimization*, 2012; 46(2):261-272.
 49. R. Sun, D. Liu, T. Xu, H. Zhang, W. Zuo. New Adaptive Technique of Kirsch Method for Structural Reanalysis. *AIAA Journal*, 2014; 52(3):486-495.
 50. Hu Wang, Yang Zeng, Enying Li, et al. "Seen Is Solution" a CAD/CAE integrated parallel reanalysis design system. *Computer Methods in Applied Mechanics and Engineering*, 2016:187-214.
 51. U. Kirsch. Implementation of combined approximation in structural optimization. *Computers and Structures* 2000; 78:449-457.
 52. K. Cai, Z. Gao, J. Shi. Compliance optimization of a continuum with bimodulus material under multiple load cases. *Computer-Aided Design* 2013; 45:195-203.
 53. R. F. Riesenfelda. On Chaikin's algorithm. *Computer Graphics and Image Processing* 1975, 4(3): 304-310.
 54. O. Sorkine, M. Alexa. As-rigid-as-possible surface modeling. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP'07)*, 2007, pp. 109-116.
 55. D. R. Jones, M. Schonlau, W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 1998; 13:455-492.