Optimizing Parameterized CAD Geometries Using Sensitivities Based on Adjoint Functions

Trevor T. Robinson[1], Cecil G. Armstrong[2], Hung Soon Chua[3], Carsten Othmer[4] and Thorsten Grahs[5]

[1] Queen's University of Belfast, t.robinson@qub.ac.uk
[2] Queen's University of Belfast, c.armstrong@qub.ac.uk
[3] Queen's University of Belfast, hchua02@qub.ac.uk
[4] Technical University of Braunschweig, c.othmer@tu-bs.de
[5] Technical University of Braunschweig, t.grahs@tu-bs.de

ABSTRACT

An approach is presented for determining which parameters defining the features in a CAD model need to be modified, and by what amount, to optimize component performance. It uses sensitivities computed for the parameters to determine the change required in each to optimize the component. Parametric sensitivity is computed by combining a measure of boundary movement due to a parameter perturbation, known as design velocity, and an adjoint sensitivity map over the boundary. The sensitivity map results from an adjoint analysis and approximates the change in objective function (performance) due to a movement of the boundary. Gradient based optimization is used based on the parametric sensitivities.

This presented method is significantly less computationally expensive than alternative approaches, and has the advantage that optimization is based on the parameters defining the CAD model, allowing it to be integrated into design workflows. The efficiency of the approach allows all of the parameters in the CAD model to be used as optimization variables, potentially offering better optimization. The work is immune to many of the issues hampering existing approaches.

Keywords: sensitivity analysis, adjoint methods, design velocity, shape optimization.
DOI: 10.3722/cadaps.2012.253-268

1 INTRODUCTION

The goal of a fully integrated design process requires that design decisions are made and implemented using computational tools managed within a Product Lifecycle Management (PLM) system. Many current approaches to design optimization are based on changing the shape or topology of the model independently of its CAD representation [3,10], or base the optimization on meshed representations of the model [15]. With these approaches shape changes cannot easily be fed back into the design workflow, and therefore do not fit within the goal of a fully integrated design process.

In modern CAD systems parameters control the dimensions of the features which make up the shape of the model. In [21] the boundary representation (BRep) control points are used as design parameters. Another method of implementing the optimization within a fully integrated design process is to use the CAD parameters as optimization variables. CAD models can be defined by a large number of parameters, meaning that this approach is not widely used, because determining the magnitude of change required in each to improve performance is extremely difficult.

A typical approach for determining the effect of each parameter is the finite difference approach [6, 16] where an analysis is run for the initial unperturbed model, and a further analysis run after the perturbation of each parameter variable in turn. The difference in the perturbed component performance and the unperturbed performance is the sensitivity. However for a model with *n* design variables this approach requires *n+1* analyses [19], where each analysis will require the CAD model geometry to be prepared for analysis, and a mesh generated. For simple models and analyses this is computationally expensive. For complex models with thousands of parameters which require manual geometry clean-up, this is unfeasible.

In an attempt to simplify the task of using parameters in a CAD model as optimization variables it is common for designers to use their experience and judgment to select, or specifically create, a small number of parameters which are deemed to be the most useful for optimizing the component. The downside of this is that it inherently reduces the design space of the optimization, limiting the amount of improvement possible. It is also possible the designer may not have a sufficient experience and engineering judgment to determine which parameters in the model are the most suitable optimization variables, which may result in parameters which have a considerable effect on performance not being selected. In general, to achieve a good improvement in performance, the number of parameters used as optimization variables should be kept large.

An efficient approach to optimization when there are a large number of design variables is to use adjoint analysis [12] which computes the sensitivity of an objective function to a small movement of the model boundary, e.g. to a surface mesh nodal position. The efficiency of the adjoint approach is that after the initial analysis of the unperturbed component only one further analysis is required to compute the adjoint sensitivity map, for a single set of optimization constraints. More information on this is given in section 3. The results can then be used to predict the change in objective function due to any number of shape perturbations without the need for more analysis.

Although adjoint methods have been used for optimization for many years, the focus of this paper is to improve performance using sensitivity values calculated by linking adjoint sensitivity maps with the boundary movements caused by perturbing the parameters defining the features in CAD models [2]. Combining the adjoint sensitivity map with the movement of the boundary provides an efficient approach to optimization, where only two analyses are required for adjoint optimization regardless of the number of parameters (for each set of constraints). This not only reduces the time required for the analysis, but also the associated geometry clean-up and meshing tasks are significantly reduced. As only one adjoint analysis is required to compute the sensitivity map, the main computational expense in the proposed approach is computing the boundary movement caused by perturbing each parameter. This means that providing the boundary movement can be computed efficiently, the number of parameters in the CAD models will not have a significant impact on the expense of the optimization.

The motivation of this paper is to present a procedure which allows efficient optimization to be based on all of the feature parameters in the CAD model, and which is applicable to any type of model in most feature based CAD systems. Existing approaches are limited to parameter perturbations which do not cause the model boundary topology to change, or analytical approaches which are only available for certain feature types. The key to this approach is the computation of boundary movement which is measured using “design velocity”. This work utilizes faceted representations of the original and perturbed geometries for this task.

Parametric sensitivity is the rate of change of the objective function (i.e. the improvement in performance) with respect to small changes in the parameter values. Parametric sensitivities are computed by combining the values in the sensitivity map with the design velocity caused by the perturbation of each parameter. It should be noted that the parameter perturbation must be such that the movement of the boundary is small. Once computed the change in performance caused by a parameter perturbation is normalized with respect to a unit perturbation to allow the benefit obtained by perturbing each parameter in the model to be compared. This information allows the most important parameters for the purposes of optimization to be identified. Given the sensitivities for all of the CAD parameters it is feasible to make engineering judgments such as the best parameters to be considered in an optimization study.

The success of the described approach at improving component performance is dependent upon the design variables used in the optimization, which for the examples in this paper range from being one user selected parameter in a finite element analysis model, to industrial CAD models where hundreds of parameters in the model are considered. It is demonstrated in the paper that one of the main benefits of this work is the ability to consider all of the parameters in the CAD model as optimization variables, therefore increasing the design space for the optimization without having to modify the CAD model in any way.

2 PARAMETERIZED CAD MODELS

In modern CAD systems component models are constructed by combining a number of modeling features to create the desired shape. Each of the features in the model is defined using existing geometry in the model (e.g. a sketch or previous modeling feature), and feature parameters which specify attributes for the feature (e.g. the length or angle of a particular aspect of the feature).

An extruded boss feature for example can be created by sweeping a sketch a certain distance in the normal direction to the sketch plane. In this instance the sweep distance is a parameter. What is not obvious is that in most CAD systems a parameter can also be specified to define the position of the other end of the extruded boss relative to the sketching plane. If the designer elects not to specify a value for this parameter, some CAD systems will set the value of this parameter to $0mm$ by default. For most features there are many options which create and specify the values of parameters in the model, however due to the nature of the CAD system the designer may not be aware of their existence. As a consequence CAD models of even simple components end up being defined by many feature parameters, and complicated models are defined by thousands of parameters. For models such as these it is impossible for the designer to know the shape change that will occur due to changing each parameter in the model, especially if they are not aware of the purpose or existence of many of the parameters. The challenge is even greater if the model is to be modified by someone other than the creator of the model, which is common.

One of the benefits of the approach presented here is that every parameter in the model is considered for use as an optimization variable, without the need for the user to understand the effect of the parameter on the model. Not only does this eliminate the time required to select appropriate optimization parameters, but the design space for the optimization is kept as large as possible. It also means a less experienced designer can carry out the optimization as no skill or engineering judgment is required to select the subset of parameters to use.

In this work the values of the parameters were automatically perturbed using the Visual Basic API for CATIA V5, which gave access to all of the parameters in the model. When optimizing a parametric CAD model using the approach described here, certain parameter types are not suitable for use as optimization variables. Parameters which should be excluded are those which:

- do not have real number values (for example parameters with integer, Boolean or string values). The optimization procedures described in this work are only suitable for parameters with continuously variable values.
- are controlled by design tables or formulae based on other parameters. Parameters controlled by other parameters or formulae are part of the design intent of the CAD model and cannot be independently varied.
- cause the model not to update when perturbed. This will occur if the parameter change invalidates the model. Clearly parameter ranges which have this effect should not be used in optimization workflows if the process is to be robust.

All of the different issues which mean that a certain parameter is not suitable for use as an optimization variable are identified using the script which perturbs the parameters. The parameter type, and whether its value is controlled by another parameter are attributes of the parameter, which can be accessed through the scripting interface. Also, the fact the model does not update successfully after a parameter perturbation can be identified by a call on the scripting interface, at which point the offending parameter value can be reset to its original value, and the model recomputed.

3 ADJOINT SENSITIVITY MAPS

The approach in this paper makes use of adjoint solutions to compute parametric sensitivity. Adjoint methods are not new, and no new work is contributed to this field in this paper. Rather an approach is presented which allows adjoint results to be used to efficiently and robustly improve the performance of the components the CAD geometries represent. The sensitivity approach presented here makes use of an adjoint sensitivity map. The sensitivity map is the results from an adjoint analysis on a surface mesh of the boundary of the model. This surface mesh information can be extracted from adjoint analysis results for the 3D model. A contour plot of the sensitivity map information for an automotive air duct is shown in Fig. 1. As the sensitivity map results from an adjoint analysis, for each element in the surface mesh, the sensitivity map allows the change in objective function due to a small outward movement of the meshed boundary at its location to be predicted.

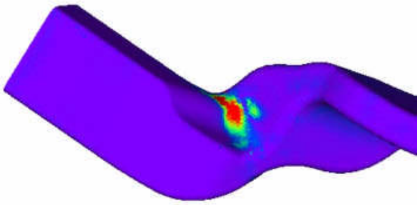

Fig. 1: An adjoint sensitivity map for a duct.

The advantage of adjoint analysis for models with a large number of design variables is that after the initial analysis only one adjoint analysis is required for each set of optimization constraints to generate the sensitivity map. For special cases where the primal and adjoint solutions are equivalent (described as being self-adjoint) the adjoint load case is not even required.

The change in objective function (*ΔJ*) caused by perturbing the value of each parameter is predicted here using the boundary method for design sensitivity described by Choi and Kim [8]. In shape sensitivity analysis the shape of the domain is changing in such a way that material can be considered as flowing in to, or out of the domain. The boundary method allows the performance measure, *J*, to be expressed in terms of the design velocity of the boundary of the model. This allows the sensitivity of performance to shape change to be described as an integral over the domain boundary of a scalar measure derived from the primal and adjoint solutions, and the normal component of design velocity.

The expressions for boundary sensitivity derived in [8] require a number of terms relating the problem to the boundary movement and external influences (for example in FEA external influences might include external loading and restraints). For special cases where the boundary of the model does not move in the region of these influences, the change in objective function (*ΔJ*) is calculated over the boundary (*A*) as

$$\Delta J = -\int_A \phi V_n dA \tag{3.1}$$

where V_n is the parametric design velocity and ϕ is the sensitivity value. Eqn (3.1) assumes the change in performance in response to a design velocity is first order, which is valid for small boundary movements which are continuous over the boundary, such as are the focus of this work. The sensitivity map in Fig. 1 shows that to reduce the objective function for the component the model boundary needs to move outwards most in the red/green regions, and that movement in the blue/purple regions will have negligible effect.

The difficulty in using the adjoint approach to optimize a CAD model is determining which parameters need to be modified to cause the boundary of the model to move outwards in the locality of the regions of highest sensitivity. Section 4 describes an approach which allows the boundary movement associated with a parameter change to be computed in a time several orders of magnitude lower than the analysis time.

4 DESIGN VELOCITY

Design velocity, V_n, is a measure of the normal displacement of the model boundary caused by an operation on the CAD model. The term parametric design velocity is used to refer specifically to the boundary movement caused by changing a parameter value. For this work each parameter was perturbed by $1mm$ which was small compared to the overall size of the model, but this was reduced automatically if any surface displacement was significantly larger than this. In Fig. 2 the design velocity at a number of points on the boundary of a CAD model due to a parameter perturbation is displayed using red arrows. The arrows represent the unit normal of the boundary scaled by magnitude of design velocity at their position.

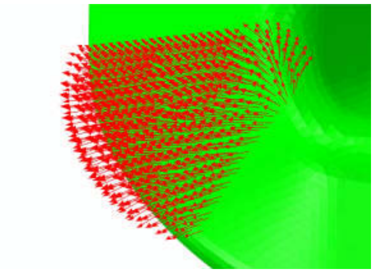

Fig. 2: Surface normal vectors scaled by design velocity.

The use of design velocity for optimization is already established, and is usually computed for meshes of the boundary [20], or directly from the geometric CAD model [13]. Choi and Chang [7] summarize and evaluate the four most common approaches to computing design velocity from the shape of the model.

Two issues hamper existing approaches to computing design velocity. The first is the 'persistent naming problem' [14], where the labels used to identify the faces in the model change during the regeneration of a CAD model. Where design velocity was determined by computing the position of a point on a face with a particular label, the wrong position is calculated if the face the label points to changes as a result of the perturbation. In Fig. 3(a) a sliver face with a particular entity label is shown shaded red. In Fig. 3(b), which shows the same region of the model after a parameter perturbation, the same entity label now relates to a different face, shown bounded in red. The corresponding original face, shown bounded in orange, has a new label.

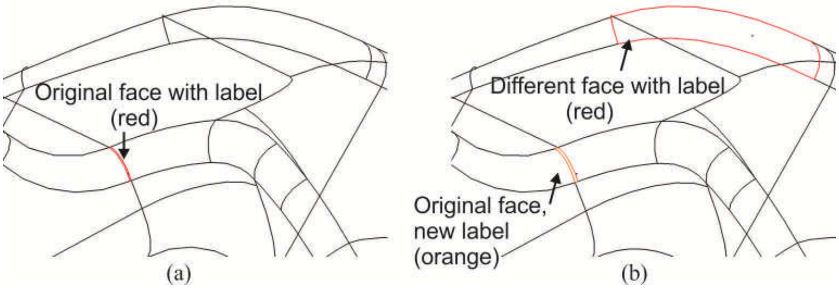

Fig. 3: Face labels changed due parameter perturbation. The shaded faces in (a) and (b) have the same ID, even though they represent different regions of the model.

The second issue is when the topology of the CAD model, or the natural mapping of a face, is affected by the perturbation of the geometry. An example of a topology change is shown in Fig. 4. Fig. 4(a) shows the unperturbed model, Fig. 4(b) shows the same region of the model after a parameter perturbation which has caused two new faces to be created, shown in blue.

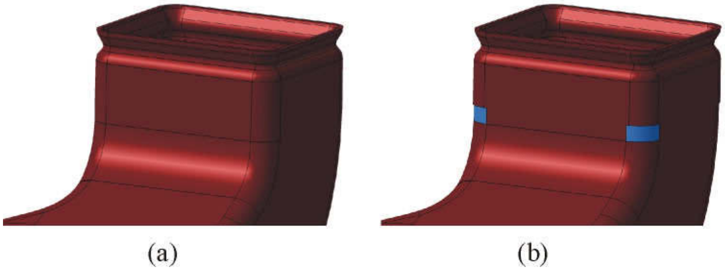

Fig. 4: Topology change due to parameter perturbation; (a) unperturbed model (b) perturbed model with new faces created.

Other geometry or meshed based approaches will only work for models where the topology of the model is unchanged after a parameter perturbation. Some researchers have elected not to work with the CAD geometry, but still make use of the CAD model parameters for optimization. Methods include determining the movement of the boundary by differentiating parametric expressions for the feature shapes based on the parameters in the construction history [1], or calculating the change in component performance due to a variation in parameter directly [5]. These approaches require access to the underlying CAD modeler kernel and are currently not available for all feature types, therefore at this time they cannot be utilized by a designer in industry. In this paper the CAD modeling system is treated as a black box, where the parameters are inputs and the model shape is the output.

The approach presented in this section for computing design velocity has proven robust and efficient. The convention used here, and in [8], is that an outward movement of the boundary is a positive design velocity and an inward movement is negative. For each point on the boundary of a component the design velocity can be computed using

$$V_n = \boldsymbol{\Delta p}.\hat{n} \tag{4.1}$$

where $\boldsymbol{\Delta p}$ is the displacement of the point on the boundary due to the parameter perturbation, and $\hat{n}$ is the outward unit normal of the surface at the point. Typically the variation in design velocity is not linear with respect to parameter variation, but in this work the size of the parameter variation is small, reducing the non-linear effects.

Working directly with CAD geometry has proven computationally expensive. This work is novel because design velocity computation was based on coarse meshes (a geometric faceting) of the original and perturbed models, which meant that topology changes and the persistent naming problem were no longer issues.

When the model was perturbed the displacement of a point on the boundary was computed by comparing the geometric facetings of the unperturbed and perturbed models. This involved projecting each bounding point in the coarse faceting of the unperturbed model onto the facets representing the perturbed component. The position of the projection point was computed by projecting the point coordinate on the unperturbed model in the direction of the surface normal of unperturbed model at that point, and determining where the projection intersected with the facets in the perturbed model. As the parameter perturbation was small the position of the closest projection point on the perturbed boundary was assumed to be the position of the original point on the perturbed boundary. The displacement at the feature bounding point due to the perturbation, *d*, was calculated as the difference between the coordinate values of the original point and its projection. This approach was carried out for the bounding vertices of all facets representing the unperturbed model.

A KD tree closest point search algorithm [11] was employed to maximize the efficiency of the computation. KD trees are efficient methods of storing data, allowing nearest neighbor searches based on coordinate values to be implemented quickly. Within the procedures, for each facet bounding point in the unperturbed model, the closest facet bounding point in the perturbed model was identified. When determining which facet the unperturbed bounding point projects onto, one of the facets bounded by the bounding point returned by the KD tree was tested first. If this projection failed a neighboring facet was tested, where the Barycentric coordinates [9] of the point on the unperturbed model relative to the facet being tested were used to determine which facet should be tested next. Neighboring facets continued to be tested in this way until a successful projection onto a facet, facet edge or vertex was found allowing the displacement to be calculated.

One further test to ensure the appropriate projection had been identified was to ensure the normal of the unperturbed boundary at the point that was projected is similar to the normal of the perturbed boundary at the identified projection point. In this work the surface normals are said to be similar when

$$\hat{n}_{original} . \hat{n}_{projection} > 0.2 \tag{4.2}$$

where $\hat{n}_{original}$ is the unit normal of the unperturbed boundary at a point, and $\hat{n}_{projection}$ is the unit normal of the boundary of the perturbed model at the projection point. This allowed the displacement to be calculated for each of the vertices bounding the facets in the unperturbed model.

If no facet was identified with a similar facet normal, or if a design velocity magnitude greater than the parameter perturbation size was computed for the closest facet with a similar facet normal, then the computation was attempted again with a reduced parameter perturbation size. In this case a new faceting was created for the newly computed geometry and the test conducted again. This was an attempt to overcome the difficulty with electing a suitable step size for this approach.

When the KD tree returned a closest point with the same coordinate values it was assumed that the boundary did not move at this point. Where three bounding points belonging to a facet are determined to have zero displacement it was assumed that the entire facet had not moved, and the design velocity for all sensitivity map elements associated with the facet were assumed to be 0.

There are a number of facet formats available, most of which can be written from the majority of CAD packages. For the work in this paper the VRML format was used. Fig. 5 shows the design velocity computed using this approach for the model where the parameter perturbation has caused the entity labels to change, Fig. 5(a) for the model shown in Fig. 3, and where the parameter perturbation has caused the model topology to change, Fig. 5(b) for the model shown in Fig.4. It is clear that there are no spurious design velocity vectors in these regions.

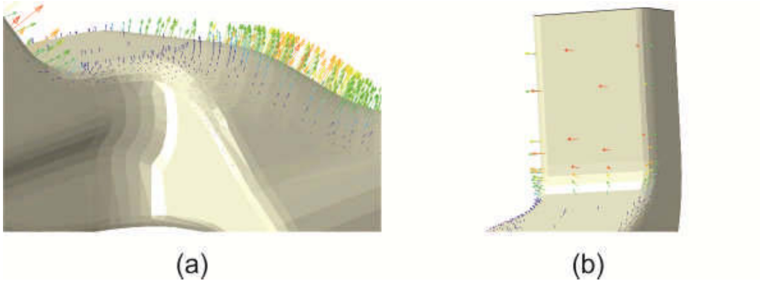

Fig. 5: Design velocity computation for models with (a) changing entity labels and (b) changing topology.

5 ASSOCIATING SENSITIVITY MAP ELEMENTS WITH THE GEOMETRIC SHAPE

As is shown in Eqn. (4.1) the dot product of the displacement at the element centroid by the element normal returns the design velocity of the element. To associate the sensitivity map elements with the geometry in order to calculate the displacement, the position of each analysis element centroid was projected onto the faceting of the unperturbed model, using the same technique described in section 4 to project the unperturbed facet bounding points onto the perturbed faceting. The parametric position of the point on the closest facet in the unperturbed model with a similar normal was recorded.

The displacement at the centroid of each sensitivity map element was efficiently computed using the displacement of the corner nodes of the coarse mesh, *d*, calculated in section 4, and the parametric position of the point's projection on the geometric facet from the unperturbed model. The displacement of the centroid (*Δp*) was computed using

$$\Delta p = u(d_2 - d_1) + v(d_3 - d_1) \tag{5.1}$$

where *u* and *v* are the parametric coordinates of the element centroid calculated for the unperturbed facet, and d_1, d_2 and d_3 are the displacements of the vertices bounding the facets, calculated by subtracting the *x, y* and *z* coordinates of the point on the boundary of the unperturbed model from the closest point on the boundary of the perturbed model. The fact design velocity is required at the element centroids for numerical integration, and away from the edges and vertices bounding a surface, means that the non-continuous design velocity measurements which may occur in these regions will not affect the approach.

Fig. 6 shows the computation of the displacement at the centroid of the sensitivity map elements. The sensitivity map element is shown as the small green element at the center of the diagram, with the parametric positions of its centroid on the blue unperturbed facet defined by *u* and *v*. Displacement was computed based on the projections of the bounding points of the associated coarse facet representing the unperturbed model, shown in blue, onto the perturbed model facets. The only perturbed facet shown (in red) is that which point d3 of the unperturbed facet projects onto.

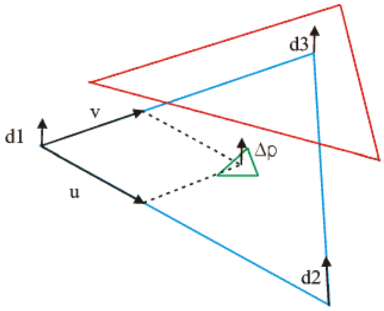

Fig. 6: The displacement of the centroid of the sensitivity map elements is based on the projections of the bounding points of the associated coarse facet.

6 PARAMETRIC SENSITIVITY

Parametric sensitivity is the change in objective function due to the perturbation of a CAD parameter, and it allows the most important parameters for the purposes of optimization to be identified. The change in objective function (*ΔJ*) caused by perturbing the value of each parameter is computed over the boundary (*A*) using Eqn. (3.1).

For this work the adjoint analysis results were provided as a mesh over the boundary of the model, so the change in objective function is summed over the surface mesh, as shown in the quadrature approximation

$$\Delta J = -\sum_{j=0}^{n} \int_{A_j} \phi V_n dA \tag{6.1}$$

The integral over each element in the sensitivity map can be evaluated numerically using

$$\Delta J = -\sum_{j=0}^{n\,elements} \sum_{k=1}^{n\,samples} w_k \phi_{k_j} V_{n_{k_j}} \tag{6.2}$$

where w_k is the weighting factor, and $\phi_k V_{n_k}$ is the product of the surface sensitivity and design velocity at *k* sampling points within each element, *j*. In this work one sampling point at the centroid of each element was used, therefore the weighting factor was A_j (element area).

For each element in the sensitivity map:

- the sensitivity at the element centroid ϕ is provided by the adjoint analysis.
- the unit area A_j and normal $\hat{n}$ at the centroid of each element in the sensitivity map can be computed using geometric considerations.
- The displacement at each element centroid (Δp) can be computed for a parameter perturbation using the procedures described in sections 4 and 5.

The change in objective function due to the parameter change is given by

$$\Delta J = -\sum_{j=0}^{n} \phi_j (\Delta p.\hat{n}) A_j \tag{6.3}$$

To compare the change in performance achieved by perturbing different parameters, the parametric sensitivity S is computed as the change in objective function due to the small change in parameter value ΔP, for which the design velocity was computed

$$S = \frac{\Delta J}{\Delta P} \tag{6.4}$$

7 EXAMPLE PARAMETRIC SENSITIVITY CALCULATIONS

Parametric sensitivities were calculated for three different examples to demonstrate the generic application of the approach. In examples 1 and 2 the objective function is strain energy which has the advantage that as it is self adjoint (i.e. the adjoint and primal solution are equivalent), an additional adjoint load case is not required, and the surface sensitivity is strain energy density over the surface. In example 1 the change in objective function is computed analytically for a cantilever beam for which one parameter controls the position of the top edge of the beam. Example 2 is a parametric 2D CAD model of the cantilever beam used in example 1, with FEA used to compute the sensitivity map.

In example 3 sensitivities are calculated for a parametric 3D duct of complex geometry. As the focus of this work is the ability to robustly use the CAD parameters for optimization, and not the adjoint solution, details of the adjoint computation are not included for this example.

7.1 Example 1: Analytical Consideration of Strain Energy in a 2D Beam

Consider the shape optimization of a plane stress model of a cantilever beam under the action of an end load, Fig. 7. The initial design was a cantilever of constant depth and was created as a simple boundary representation with four bounding edges. The length of the beam (L) and the depth at the fixed end (d) are the dimensional parameters in the model. Geometric constraints ensured the bottom edge was kept horizontal, and the left and right edges were kept vertical. The length was assumed fixed so the only free variable was the depth at the fixed end. The beam was subject to a constraint of constant mass so that when the depth of the left hand side increased, the depth of the right hand side decreased by the same amount.

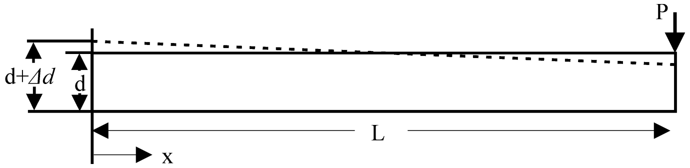

Fig. 7: A parametric cantilever beam, with one parameter, *d*, controlling the depth of the fixed end.

As strain energy was the objective function (J), the sensitivity function (ϕ) was the strain energy density, given by

$$\phi = \frac{\sigma^2}{2E} \tag{7.1}$$

The bending stress, σ, along the top edge of the model is given by

$$\sigma = \frac{My}{I} = \frac{12P(L-x)y}{bd^3} = \frac{6P(L-x)}{bd^2} \tag{7.2}$$

where *I* is second moment of area and *y = d/2*. Substituting Eqn. (7.2) into Eqn. (7.1) results in

$$\phi = \frac{18P^2(L-x)^2}{b^2Ed^4} \tag{7.3}$$

The beam was constrained in such a way that perturbing the parameter (*d*) by a small amount (*Δd*) only caused design velocity along the top edge. The consequence of the constant mass constraint being imposed on the beam was that the design velocity along the top edge could be expressed in terms of the parameter perturbation *Δd*, as

$$V_n = \frac{\Delta d}{L}(L-2x) \tag{7.4}$$

Substituting the expressions for ϕ and V_n into Eqn. (3.1), the change in objective function (*ΔJ*) due to a perturbation in the value of parameter *d* by *Δd* is therefore given by

$$\Delta J = -\int_0^L \frac{18P^2(L-x)^2}{b^2Ed^4}\frac{\Delta d}{L}(L-2x)dx \tag{7.5}$$

Integrating Eqn. (7.5) gives an expression for the change in objective function due to a perturbation in the parameter by *Δd*. The parametric sensitivity was calculated as

$$S = \frac{\Delta J}{\Delta d} = -\frac{3P^2L^3}{b^2d^4E} \tag{7.6}$$

7.2 Example 2: 2D Stress Consideration of a Beam

The cantilever beam shown in Fig. 7 was modeled and analyzed using the ABAQUS finite element analysis package. A structured mesh comprised of 8 noded quadrilateral elements was used for the analysis, with 80 elements evenly distributed along the length of the beam and 8 through the depth. The analysis attributes were: *L=10, P=1, b=1, d=1, E = 200000.*

The unperturbed model (*Δd=0*) was analyzed to establish the initial strain energy density distribution along the top edge of the beam. As only the top edge of the cantilever beam was displaced by the perturbation, the sensitivity map consisted of the element edges on the top edge of the beam. The strain energy density values at the mid-point nodes on the top edge of the model were recorded (80 values), and used as the sensitivity values for the sensitivity map for the analysis.

The parameter controlling the depth of the fixed end was perturbed by *Δd*. Due to the nature of the mesh the *x* coordinate values for the mid side nodes remained constant, and the change in *y* coordinate represented the design velocity for each sensitivity map element. This was because the edge normal was parallel to the *y* axis in the unperturbed model.

The change in objective function was computed as the summation for each element of the product of:

- length of the unperturbed element edge on the boundary (in this example 0.125)
- design velocity for each mid-side node (represented by the change in y coordinate)
- strain energy density computed at the mid-side node for the unperturbed model

The summation of the *ΔJ* values for each element was the change in objective function for the beam due to a perturbation of parameter *d* by *Δd*. The change in objective function was calculated for 5 different magnitudes of perturbation (*Δd* = 0.005, 0.01, 0.025, 0.05 and 0.1).

Fig. 8 compares the change in objective function due to increasing the depth at the fixed end of the beam with the magnitude of the parameter perturbation. The datasets represent:

i. ΔJ Analytical - the analytical result derived in section 7.1, calculated using Eqn. (7.5).
ii. ΔJ FEA - a finite difference calculation which involved running a finite element analysis for each perturbed model, and subtracting the resulting strain energy for the perturbed model from the value computed for the nominal model.
iii. ΔJ Sensitivity - the parametric sensitivity approach, calculated using Eqn. (6.3).

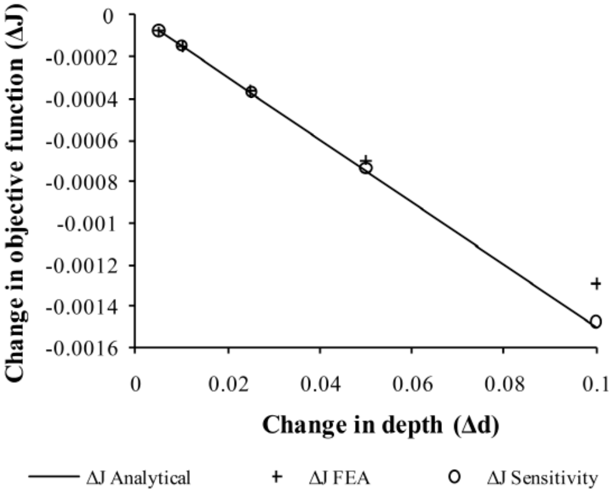

Fig. 8: Change in objective function with perturbation size for the analytical, sensitivity and finite element analysis considerations of a parametric cantilever beam.

The sensitivity calculated from the design velocity and strain energy on the boundary accurately predict the change in performance for a small change in shape. There is a difference of 1.7% between the value calculated using the sensitivity approach and the analytical approach. As the size of the parameter perturbation becomes dramatically different to the value which was used to compute the parametric sensitivity value, the FEA results diverge from the analytical/sensitivity predictions because the nonlinear relationship between the design velocity and the parameter perturbation begins to have an effect. It is worth noting that the sensitivity method required one analysis in total, whereas the conventional FEA required one analysis for the unperturbed model, and one for each perturbation size (i.e. 6 in total).

7.3 Example 3: 3D CFD Consideration of an Air Duct

Parametric sensitivities were computed for an industrial model of an automotive air duct, a region of which is shown in Fig. 1 along with its sensitivity map. The sensitivity map used for this model was computed using OpenFOAM [17] and an in-house adjoint analysis code the details of which are outside the scope of this paper [18, 19]. It was provided as an OpenFOAM boundary mesh, with associated values representing the sensitivity value at the centroid of each element. The red regions of the sensitivity map represent regions where an outward movement of the boundary would reduce the objective function. As the sensitivity information is provided at the centroid of the element the change in objective function can be computed using Eqn. (6.3).

A Visual Basic Script was written to implement the entire process and automatically compute the parametric sensitivity for each parameter in the model. The design velocity was calculated as the inner product of the displacement vector and the face normal of the OpenFOAM element.

The model shown in Fig. 1 is defined in CATIA V5 using 1015 parameters of which 174 have real values and are suitable for use in a sensitivity analysis. In practice optimization of this model would possibly be attempted using only 5 manually selected parameters. One conventional CFD analysis of the duct had a computation time of about 3 hours on 8 CPUs, meaning a traditional finite difference approach, requiring an individual analysis for each parameter perturbation would have required in excess of 15 hours for the 5 selected parameters, and in excess of 3048 hours (over 2 weeks) for the 174 parameters.

Using the sensitivity approach described in this paper, the initial analysis required 3 hours, the adjoint analysis required 3 hours computation time, and the calculation of design velocity and the parametric sensitivities for all parameters in the model required 41 minutes. Here the CFD analysis was carried out on a cluster with 8 CPUs whereas the computation of parametric sensitivities for all 174 real value parameters (not just the selected parameters), were computed in 41 minutes (of which the design velocity computation required 6 minutes) on a standard Intel Pentium 4 workstation with 3GB of memory.

These values indicate that the approach presented can be used to calculate parametric sensitivities for all of the parameters in the model quicker than the finite difference approach can for the small number of selected parameters. Not only is this approach more efficient, it has the potential to provide a much better performing component as more design variables can feasibly be used, and some may have higher sensitivity values than designer selected parameters. There is the opportunity to run the design velocity computation, and to associate the sensitivity map elements to the coarse faceting in parallel with the computational or adjoint analysis. This would mean the time required to compute the design velocity and the mapping will not impact on the parametric sensitivity computation time, providing it is less than the CFD+adjoint analysis times. When the adjoint analysis is complete the sensitivity values for each sensitivity map element can be combined with the design velocity to compute parametric sensitivities in a very short time.

8 COMPUTING PARAMETRIC CHANGES

Once parametric sensitivities have been computed for all parameters in the model, they can be used to determine the optimum perturbation size to apply to each in order to produce the optimum change in shape. The change in objective function due to small change in the value of any parameter can be computed using

$$\Delta J = S_1\Delta P_1 + S_2\Delta P_2 + ... = \sum_{i=1}^{n\ parameters} S_i \Delta P_i \tag{8.1}$$

Gradient based optimization approaches make use of sensitivity values to calculate changes in design variables. In this work, to achieve the optimum movement of the boundary, the perturbations applied to each parameter were proportional to the parametric sensitivities calculated for each, (i.e. in the direction of steepest descent, [4]). The vector of steepest descent in parameter space is computed as

$$\Delta P = -x\{S_1 \quad S_2 \quad .....\} \tag{8.2}$$

where x is a multiplier applied to the parametric sensitivities calculated for each parameter. The change in objective function due to changing the parameter values using the vector in Eqn. (8.2) is

$$\Delta J = -x({S_1}^2 + {S_2}^2 + ...) = -x\sum_{i=0}^{n} {S_i}^2 \tag{8.3}$$

One of the goals of this work was to predict the value of x required to achieve a given change in performance. Shape sensitivity is accurate for small parameter perturbations, so the objective here is to calculate a parameter perturbation which keeps the predicted change in objective function small. The multiplier x to achieve a fractional change in performance $\varepsilon\Delta J$ can be estimated using

$$x = -\frac{\varepsilon\Delta J}{\sum_{i=0}^{n} {S_i}^2} \tag{8.4}$$

Such a vector was computed to perturb the selected parameters for the air duct shown in example 3. Fig. 9 shows the duct before (in blue dashed lines) and after (in red solid lines) the parameters have been perturbed in the steepest descent direction. In regions where no displacement has occurred the blue lines are hidden behind the red and are not visible. Notice that the greatest displacement of the boundary is localized in the regions which are shaded red/green in the sensitivity map in Fig. 1. This demonstrates that this vector has successfully caused the boundary to move most in the areas of highest sensitivity.

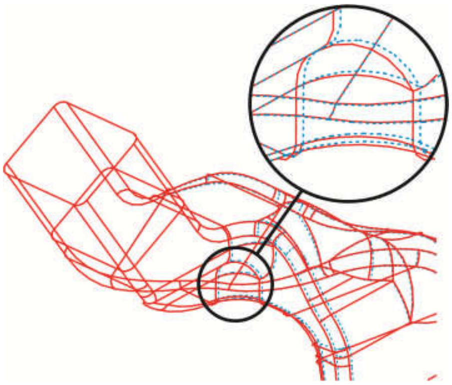

Fig. 9: The air duct before (in blue dashed lines) and after (in red lines) the parameters have been perturbed along the steepest descent vector.

Fig. 10 shows the correlation for an air duct, and compares the change in objective function predicted using the procedures in this paper to the change in objective function computed by a CFD analysis for:

i. a perturbation of different amounts in the positive and negative directions along the steepest descent vector

ii. a perturbation of the two parameters of highest predicted sensitivity to give no change in performance. For this example one parameter was perturbed by an amount predicted to give a certain change in performance, and the other parameter was perturbed by an amount predicted to give the opposite change in performance.

iii. Predicted and measured change in performance when selected individual parameters are changed by 1*mm*.

The trend line in Fig. 10 shows the linear correlation between the predicted and actual results.

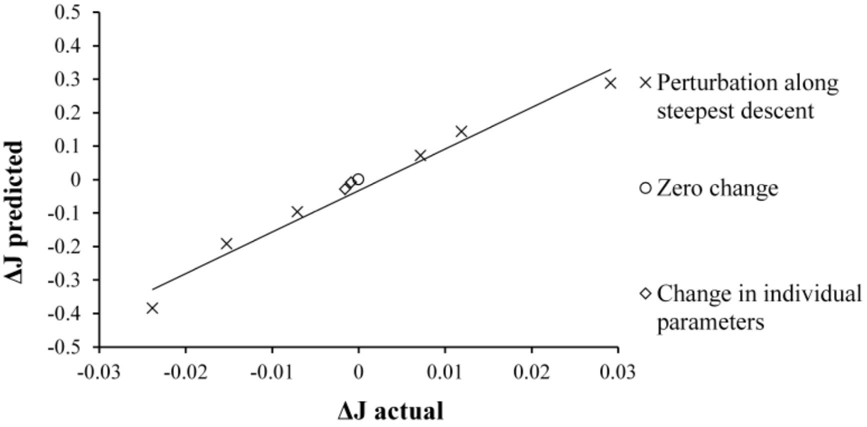

Fig. 10: The change in objective function predicted for an air duct against the change in objective function computed by a CFD analysis using the steepest descent vector and individual parameters.

Through the course of this work it became apparent that the adjoint sensitivity map provided by the CFD computation was out by a factor of around 12. This was identified by this work, and was subsequently proven using a number of methods. Fig. 10 shows the good correlation between the performance change predicted in an automated way using the approach outlined in this paper, and the performance change calculated using CFD finite differences.

9 DISCUSSION

The primary objective of this work was to automate the process of optimizing parameterized CAD models using of all of the real value parameters in the CAD model as optimization variables, and not just a subset of the parameters as is attempted in other research. This includes parameters created with the feature by default, with no input from the designer. The approach presented is applicable to any parameterized CAD model, and can be used with any feature based modeling system capable of writing facetted model approximations. Significant advances have been made as a result of this work.

It is the efficient and robust computation of design velocity for any real value parameter in a CAD model that is the primary benefit of this work. It is efficient because the geometric faceting to map the design velocity over the boundary can be significantly coarser than the analysis mesh used for the sensitivity map, allowing fewer projections to be used to compute surface displacement. Once the analysis mesh has been associated with the geometric faceting of the original model, the design velocity at any given analysis mesh element can be computed efficiently. The approach presented is robust for even complex geometries, and is unaffected by the topology of the model changing and the persistent naming problem which cause other approaches to fail.

Since the contribution to shape sensitivity is given by Eqn. (3.1), the required accuracy at the geometric faceting used to compute V_n depends on the values of the sensitivities on a given surface. In areas where the sensitivity is large a denser faceting to resolve V_n more accurately is required. Settings within the CAD system allow the facet density to be altered if required.

Example 3 has demonstrated the significant reduction in analysis time achieved using the approach suggested here, therefore allowing all parameters to be considered. Running the procedures described for all real value parameters, as opposed to just a subset, allows the most significant ones for optimization to be identified. This is not feasible using other approaches due to the expense and difficulty of computing the sensitivity for each CAD parameter using finite differences and multiple CFD analyses. Not only does considering all parameters have the potential to increase the amount by which the performance of the model can be improved, it saves the designer having to identify the parameters likely to be the most useful for optimization when preparing the CAD model. This will reduce the model preparation time and the knowledge required by the designer. The approach can be applied to any CAD model and any type of feature, including complex constructions and blends such as can be seen in Fig. 1.

Once sensitivities have been calculated for all parameters a more efficient optimization process can be achieved by only perturbing the parameters which have the highest parametric sensitivity values.

Predictions based on the sensitivity map are not valid for large perturbations in shape. In practice if a suitable reduction in objective function is not obtained after a first iteration, where parameters are perturbed by a small amount, a new sensitivity map should be created for the perturbed geometry by carrying out another analysis. Although this adds expense it is necessary to be able to compute changes in performance accurately. Using this approach the optimum parameter values should be identified after a number of iterations.

The example models demonstrate a good correlation between the predicted value for change in objective function using the sensitivity approach presented here, and the change in objective function computed analytically or using computational analysis. Also demonstrated is the use of parametric sensitivity values to perturb the model in such a way as to produce no change in performance.

10 CONCLUSIONS

From this work it can be concluded that:

- The approach can be used to calculate parametric sensitivities for CAD models in significantly less time and with a lower computational requirement than traditional approaches.
- Coarse facetings of the geometries and the KD tree enable the efficient computation of design velocity and parametric sensitivities even when the topology of the boundary of the CAD model changes.

- A vector in parameter space can be computed of the parametric sensitivity values for all of the parameters. This can be used to simultaneously perturb all of the parameters in the model to achieve the greatest performance improvement.
- The parametric sensitivities can be used to choose the best parameters for a detailed optimization study.

11 REFERENCES

[1] Alonso, J.; Martins, J.; Reuther, J.; Haimes, R.; Crawford, C.: Aero-Structural Design Using a Parametric CAD-Based Model", Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA 2003-3429. Orlando FL, June 2003.

[2] Armstrong, C.G.; Robinson, T.T.; Ou, H.; Othmer, C.: Linking adjoint sensitivity maps with CAD parameters, Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, P. Neittaanmaki, J Periaux and T Tuovinen (Eds.), 234-239, Eurogen 2007, Jyväskylä, Finland, 2007 (ISBN: 978-951-39-2863-6).

[3] Bendsøe, M.P.; Rodrigues, H.C.: Integrated topology and boundary shape optimization of 2-D solids, Computer Methods in Applied Mechanics and Engineering, Volume 87, Issue 1, May 1991, Pages 15-34, DOI: 10.1016/0045-7825(91)90144-U

[4] Bertsekas, D.P.: Nonlinear programming: Second edition, Athena Scientific, Belmont, Massachusetts, 2003

[5] Chen, J.; Freytag, M.; Shapiro, V.: Shape Sensitivity of Constructively Represented Geometric Models, Computer Aided Geometric Design, 25(7), 470-488 2008 DOI: 10.1016/j.cagd.2008.01.005

[6] Chen, S.; Tortorelli, D.A.: Three-dimensional shape optimization with variational geometry, Structural Optimization, 13, 81-94, 1997. DOI: 10.1007/BF01199226

[7] Choi, K.K.; Chang, K.-H.: A study of design velocity field computation for shape optimal design, Finite Elements in Analysis and Design, 15(4), 317-341, 1994, DOI: 10.1016/0168-874X(94)90025-6

[8] Choi, K.K.; Kim, N.-H.: Structural Sensitivity Analysis and Optimization 1, Springer, New York, 2005.

[9] Ericson, C.: Real-Time Collision Detection, Amsterdam; London: Morgan Kaufmann, 2005

[10] Fernandes, P.; Guedes, J.M.; Rodrigues, H.: Topology optimization of three-dimensional linear elastic structures with a constraint on "perimeter" Computers & Structures, Volume 73, Issue 6, December 1999, Pages 583-594 DOI: 10.1016/S0045-7949(98)00312-5

[11] Freidman, J.H.; Bentley, J.L.; Finkel, R.A.: An Algorithm for Finding Best Matches in Logarithmic Expected Time, ACM Transactions on Mathematical Software (TOMS), v.3 n.3, p.209-226, Sept. 1977 DOI: 10.1145/355744.355745

[12] Giles, M.B.; Pierce, N.A.: An introduction to the adjoint approach to design, Flow Turbul. Combust., 65, 393–415, 2000.

[13] Hardee, E.; Chang, K.-H.; Tu, J.; Choi, K.K.; Grindeanua, I.; Yu, X.: A CAD-based design parameterization for shape optimization of elastic solids, Advances in Engineering Software, 30, 185–199, 1999. DOI: 10.1016/S0965-9978(98)00065-9

[14] Kripac, J.: A mechanism for persistently naming topological entities in history-based parametric solid models, Proceedings of the Third ACM Symposium on Solid Modeling and Applications, 21–30, 1995.

[15] Lindby, T.; Santos, J.L.T.: 2-D and 3-D shape optimization using mesh velocities to integrate analytical sensitivities with associative CAD, Structural and Multidisciplinary Optimization, Volume 13, Number 4, 213-222, DOI: 10.1007/BF01197449

[16] Lindby, T.; Santos, J.L.T.: Shape optimization of three-dimensional shell structures with the shape parameterization of a CAD system, Structural and Multidisciplinary Optimization, 18 (2-3), 126-133, 1999. DOI: 10.1007/BF01195987

[17] OpenFOAM: http://www.openfoam.com/, Accessed: 08/06/10

[18] Othmer, C.: A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows, Int. J. Num. Meth Fluids, 58(8), 861–77, 2008. DOI: 10.1002/fld.1770

[19] Othmer, C.; Grahs, T.: CFD topology and shape optimization with adjoint methods, VDI Fahrzeug- und Verkehrstechnik, 13. Internationaler Kongress Berechnung und Simulation im Fahrzeugbau, Würzburg, VDI-Berichte, 1967, 61-72, 2006.

[20] Toivanen, J.I.; Martikainen, J.: A new method for creating sparse design velocity fields, Comput. Methods Appl. Mech. Engrg. 196, 528–537, 2006. DOI: 10.1016/j.cma.2006.05.001

[21] Yu, G.; Müller, J.-D.; Jones, D.; Christakopoulos, F.: CAD-based shape optimisation using adjoint sensitivities, Computers & Fluids. 46, 512-516, 2011.