Available online at www.sciencedirect.com

SCIENCE DIRECT®

Comput. Methods Appl. Mech. Engrg. 194 (2005) 3213–3243

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Review of options for structural design sensitivity analysis. Part 1: Linear systems

F. van Keulen [a,*], R.T. Haftka [b], N.H. Kim [b]

[a] Koiter Institute Delft, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands

[b] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, FL 32611, United States

Abstract

The paper reviews options for structural design sensitivity analysis, including global finite differences, continuum derivatives, discrete derivatives, and computational or automated differentiation. The objective is to put these different approaches to design sensitivity analysis in the context of accuracy and consistency, computational cost, and implementation options and effort. Linear static analysis and transient dynamic analysis are reviewed. In a separate appendix, special attention is paid to the semi-analytical method. A future paper will address design sensitivity analysis in nonlinear structural problems.

Keywords: Design sensitivities; Structural response derivatives; Shape design variables; Linear statics; Transient analysis; Review

1. Introduction

There are currently four broad categories of methods in common use for obtaining the derivatives of performance measures with respect to structural parameters: (a) overall finite differences; (b) discrete derivatives; (c) continuum derivatives; and (d) computational or automatic differentiation. The differences between these methods are particularly important for shape sensitivities, because shape design variables change the discretization, i.e. mesh or grid, when numerical techniques such as finite element, boundary element, or meshfree methods are used. In this paper, the sensitivity is defined as a derivative of a performance measure with respect to a design variable.

Except for the finite differences option, the other three come in direct and adjoint methods (called the reverse mode for automatic differentiation). In the direct mode, one obtains the derivatives of the entire

[*] Corresponding author.

0045-7825/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2005.02.002

structural response, and often of intermediate quantities as well. The sensitivities of performance measures can then be obtained from the chain rule of differentiation. In the adjoint mode, one defines an adjoint problem, which depends on the performance measure. The sensitivities of performance measures can then be obtained using the structural and adjoint responses. Thus not all system response sensitivities are required, which is particularly advantageous in cases with many design variables, but few performance measures of interest.

One criterion for choosing a class of methods is accuracy. Clearly, accuracy of the sensitivities may influence the optimization solution, the required number of optimization cycles and premature convergence. Unfortunately, accuracy is subjective, and here we differentiate between accuracy and consistency. We define *accuracy* to be the difference between the derivatives we obtain and the exact derivatives obtained from an exact solution of the governing continuum equations. Obviously such an exact solution is not known for most practical problems and can only be approximated by sufficiently accurate numerical models. We define *consistency* as the difference between the derivatives we obtain and the exact derivatives of the numerical model. Thus, in the context of consistency the accuracy of the underlying (numerical) model used for the response functions is irrelevant. The difference between consistency and accuracy is due to the fact that the accuracy of the numerical solution changes over the design space, and different discretizations may be used for evaluating response functions and derivatives. These definitions are illustrated in Fig. 1 for a noiseless response (a), as well as a noisy response (b). This noise could, for example, be the result of remeshing.

The choice between the different options for calculating derivatives is also influenced by two other criteria: *computational cost* and *implementation effort*. The objective of the present paper is to focus on papers that provide guidance on the choice of methods based on the three criteria: accuracy and consistency, computational costs and implementation efforts.

Design sensitivity analysis has been addressed in other survey papers. We first refer to the review by Haftka and Adelman [1], which is already somewhat dated. Kwak [2] and Hsu [3] summarize available techniques in computational shape optimization. The review of Ref. [4] is focused on aerodynamic optimization and related (multidisciplinary) complexities. Kleijnen [5] describes the use of sensitivities in a broad sense.

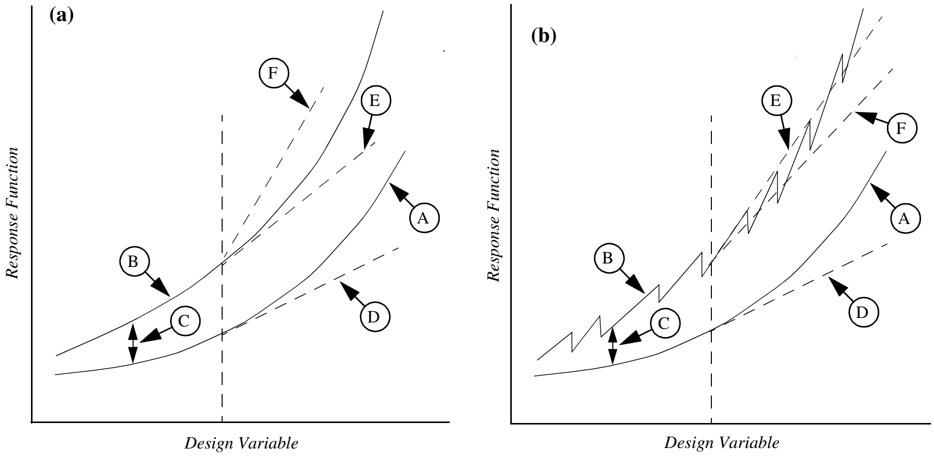

Fig. 1. Sketch of response and different sensitivities for a nearly noisefree simulation (a) and a noisy (possibly due to remeshing) simulation (b). The curves labeled "A" correspond to the exact solution of the governing continuum equations. The computational counterparts are denoted "B". The differences between these curves is the modeling error, denoted "C". Exact derivatives are identified with "D". Consistent derivatives are labeled "E", whereas non-exact and non-consistent COMPUTED derivatives are denoted by "F".

In order to achieve a maximum level of coherence we limit ourselves to linear static and transient analyses. One important area not discussed is the sensitivity of bifurcation, limit loads, and vibration problems that lead to eigenvalue problems. The methods used for calculating sensitivity of eigenvalues are quite specialized and are difficult to integrate in this paper. The interested reader may refer to Refs. [6,7].

Other areas not discussed herein include coupled multidisciplinary problems, such as system sensitivity using Sobieski's global sensitivity equations [8–11], fracture [12–14], and structural-acoustics [15–17].

Section 2 introduces the four methods herein. Section 3 compares them for linear static analysis, while Section 4 compares them for transient analysis. The semi-analytical method is discussed briefly in Section 3 and Appendix A provides much more detail. Finally, in Appendix B an index to the references in this paper is given.

The calculation of sensitivities usually requires the solution of systems of algebraic equations. Throughout the paper a direct solver is assumed unless specified otherwise. For a more detailed discussion on iterative matrix solver in sensitivity analysis, the reader is referred to Refs. [18–26].

2. Methods of sensitivity calculations

2.1. Introduction

Linear structural behavior is often assumed or invoked as an initial approximation. Linearity implies that the strain tensor $\boldsymbol{\epsilon}_c$ is given in terms of the displacement vector $\mathbf{u}_c$ as

$$\boldsymbol{\epsilon}_c = \tfrac{1}{2}\left(\operatorname{grad}\mathbf{u}_c + \operatorname{grad}^{\mathrm{T}}\mathbf{u}_c\right). \tag{1}$$

The subscript "c" is introduced to emphasize that these quantities belong to the continuum description. Linearity also implies that the stress tensor $\boldsymbol{\sigma}_c$ is given as

$$\boldsymbol{\sigma}_c = \mathbf{S}_c : \boldsymbol{\epsilon}_c, \tag{2}$$

with $\mathbf{S}_c$ being the fourth-order elasticity tensor, which may be a function of the spatial coordinates but is independent of the deformations and the stresses. Finally, the loads acting on the structure must be independent of the displacements. The principle of virtual work, convenient for formulating the equations of equilibrium, reads as

$$\int_V \delta\boldsymbol{\epsilon}_c : \mathbf{S}_c : \boldsymbol{\epsilon}_c \,\mathrm{d}V = \int_V \rho\mathbf{b}_c \cdot \delta\mathbf{u}_c \,\mathrm{d}V + \int_A \mathbf{h}_c \cdot \delta\mathbf{u}_c \,\mathrm{d}A \tag{3}$$

for all $\delta\mathbf{u}_c$ that belong to the space of kinematically admissible displacements. In (3), ρ is the mass density; $\delta\boldsymbol{\epsilon}_c$ is the virtual strain interpreted from (1) using $\delta\mathbf{u}_c$; $\mathbf{b}_c$ denotes the external load per unit mass; and $\mathbf{h}_c$ reflects tractions acting on the outer surface A of the structure.

As the above equations can be solved analytically for only few practical cases, discrete approximations are commonly formulated. With the finite element method, the discrete counterpart of (1) becomes

$$\boldsymbol{\epsilon}_e = \mathbf{D}_e\mathbf{u}_e, \tag{4}$$

where $\boldsymbol{\epsilon}_e$ denotes a set of generalized deformations for a single element and $\mathbf{u}_e$ are the element nodal degrees of freedom. Note that, the subscript "e" implies the discretized element level. Since the discussion is restricted to the linear regime, (4) can be used for the definition of $\delta\boldsymbol{\epsilon}_e$ when $\mathbf{u}_e$ are replaced by $\delta\mathbf{u}_e$. Denoting energetically conjugate generalized stresses as $\boldsymbol{\sigma}_e$, the internal virtual work for a single element (δW_e^i) is

$$\delta W_e^i = \int_{V_e} \boldsymbol{\sigma}_c : \delta\boldsymbol{\epsilon}_c \,\mathrm{d}V = \boldsymbol{\sigma}_e \cdot \delta\boldsymbol{\epsilon}_e. \tag{5}$$

The linear discrete constitutive equations are expressed as

$$\boldsymbol{\sigma}_e = \mathbf{S}_e \boldsymbol{\epsilon}_e, \tag{6}$$

where $\mathbf{S}_e$ is an elasticity matrix. Due to the discretization process and the particular definitions used for the generalized deformations and stresses, the matrix $\mathbf{S}_e$ will be a function of the dimensions of the elements as well. Summing over all elements, the principle of virtual work for the entire discretized model then yields

$$\sum_e \boldsymbol{\sigma}_e \cdot \delta\boldsymbol{\epsilon}_e = \mathbf{f} \cdot \delta\mathbf{u}, \tag{7}$$

where $\mathbf{u}$ is a system vector containing all nodal degrees of freedom and $\mathbf{f}$ is a vector of nodal loads. Note that the subscripts have been omitted for the system level. Eq. (7) must be satisfied for all $\delta\mathbf{u}$ that satisfy the homogeneous essential boundary conditions. Combining (4)–(7), the well-known result

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{8}$$

is found, where $\mathbf{K}$ is the stiffness matrix which is assembled from the individual element matrices $\mathbf{K}_e$, with

$$\mathbf{K}_e = \mathbf{D}_e^{\mathrm{T}} \mathbf{S}_e \mathbf{D}_e. \tag{9}$$

Note that the matrix $\mathbf{K}$ is symmetric and generally sparse. A boundary element method produces a linear set of equations similar to (8). For comparable accuracy, the boundary elements requires fewer equations, but $\mathbf{K}$ is not sparse and may not be symmetric.

The choice of solver for Eq. (8) affects to what extent investments made for solving (8) can be reused for design sensitivity analysis. A direct solver typically factorizes $\mathbf{K}$ taking advantage of features such as symmetry and sparsity. This factorization can easily be used for other load cases or, as will be shown later, for calculation of the design sensitivities. This can substantially reduce the computational cost of design sensitivities. Direct solvers are particularly efficient for sets of equations with a special structure, e.g. with a small profile or tightly banded, or relatively small sets of equations, or sets of equations that suffer from bad conditioning. Because iterative solvers are sensitive to conditioning, they are often combined with pre-conditioners. For design sensitivity analysis, iterative solvers have the disadvantage that investments made for solving (8) are difficult to reuse. The exception is the investments involving pre-conditioning. These efforts can normally be reused for the sensitivity analysis. Moreover, the investments for preconditioning can also be used in a global finite difference setting. In that case the preconditioner for the original configuration is used for the perturbed configurations as well.

Depending on the structure of the matrix (symmetric versus non-symmetric) and the convergence characteristics of the iterative solution procedure (linear versus super-linear), methods that ease the solution for another right-hand side vector may also be available [27–29].

2.2. Global finite differences

Overall or global finite difference consist of repeated execution of the analysis code and the use of a finite difference formula to obtain the derivative. Forward or backward differences are the most popular, the use of central differences is not uncommon, but higher order difference formulae are very rare.

Finite difference derivatives can suffer from truncation errors with large step sizes and also from errors when the step size is too small. The most obvious source of the latter type of error is computational errors associated with arithmetic involving a finite number of digits, i.e. round-off errors, and possibly ill-conditioning in the problem. Other potential sources are the discretizations of both the spatial and the temporal domains. A typical example could be numerical noise induced by remeshing (see, e.g., Fig. 1b). Finally, this type of error may be triggered by iterative processes which are stopped as soon as a certain accuracy criterion is satisfied. In the sequel we shall refer to this type of error as *noise error*.

Computational efficiency, accuracy and consistency, and implementation effort for global finite differences depend to a large extent on the type of solvers used for the linear equations (8). The main issue is whether computational investments associated with solving the equations for the nominal structure can help reduce the effort associated with their solution for a perturbed structure.

When direct solvers are used for the solution, so that the matrix **K** has been factored, there is an array of methods that provides fast re-analysis of the perturbed structure. A disadvantage of many of these techniques is that accuracy is generally compromised, i.e. certain inaccuracies will be introduced. When the perturbation leads to a low rank modification of **K**, for example because only a single finite element is modified, then an exact analysis of the perturbed structure can be performed using the Sherman–Morrison–Woodbury formulae [30]. The main computational cost of this approach is the solution of (8) for a number of right-hand sides equal to the rank of the perturbation in **K**. Akgün et al. [30] discuss several variants of this approach including the method of virtual distortions.

When the perturbation in the matrix is more extensive, as in shape variation, it is still possible to use a binomial series solution [31,32] or a similar approximation of the inverse of **K** using a Neuman series [33,34]. A more sophisticated reanalysis technique can be found in Ref. [35].

When iterative solvers are used for the solution, they are often combined with pre-conditioners. It may be possible to use the same pre-conditioner for the perturbed solution, thus reducing the computational cost of the derivative. Moreover, the nominal solution can be used as a good starting point for the iteration process associated with the perturbed configuration.

It must be emphasized, that reanalysis techniques have often been published in a different context to global finite differences. The point is that the distinct differences between global finite differences and more sophisticated discrete sensitivities fade as soon as refinements to the global finite difference schemes are introduced [36]. This is particularly so for very small design perturbations. Yet it may be advantageous to look at reanalysis from the aspect of global finite differences with full reanalysis in the loop since a full reanalysis may result in better accuracy. More sophisticated methods generally result in better computational efficiency, although they may be less accurate.

A very special form of finite differences is the one which relies on complex variables. As this method can be seen as a special form of computational derivatives, the discussion of the complex variables approach is included in the section on computational derivatives.

2.3. Continuum derivatives

Continuum derivatives are obtained by differentiating the continuum equations that govern structural behavior. Most commonly these consist of partial differential equations or an integral form, for example, derived from the principle of virtual work. The differentiation leads to a set of continuum sensitivity equations that are then solved numerically, usually with the same discretization, but not necessarily, as used for the original structural response. For shape sensitivities, the two main approaches for continuum derivatives are the material derivative approach [37–42,13] and the control volume approach [43–45].

In the continuum approach, the design variables may be considered as fields which are functions of the spatial coordinates. As a consequence, sensitivity is to be understood as a variation of a function. Let us consider that the design variable s is perturbed to $s + \tau\eta$ in which τ is the scalar that measures the perturbation size and η is the direction of design change. For simplicity, it is assumed that the structural design variable s does not affect the domain. The variation of field response $\mathbf{u}_c$ with respect to s can then be defined as

$$\mathbf{u}_c' \equiv \lim_{\tau\to 0}\left\{\frac{\mathbf{u}_c(s+\tau\eta)-\mathbf{u}_c(s)}{\tau}\right\} = \left.\frac{\partial \mathbf{u}_c}{\partial s}\right|_{\tau=0}\eta. \tag{10}$$

Since the direction of design change η can be arbitrary, (10) must be linear with respect to η and the coefficient of η is called the sensitivity of field response $\mathbf{u}_c$, which is equivalent to the derivative in the context of other approaches.

Using (10) and the linear property, the variation of continuum strain tensor in (1) can be obtained as

$$\epsilon_c' = \tfrac{1}{2}\left(\operatorname{grad}\mathbf{u}_c' + \operatorname{grad}^{\mathrm{T}}\mathbf{u}_c'\right). \tag{11}$$

This equation indicates that the strain variation has the same linear relationship to the displacement variation. Similarly, the variation of (2) yields

$$\boldsymbol{\sigma}_c' = \mathbf{S}_c' : \epsilon_c + \mathbf{S}_c : \epsilon_c'. \tag{12}$$

This equation indicates that if the constitutive equations depend on the design variables, the derivative equation will have an additional "initial-strain-like" loading. This case covers not only design variables that control material properties, but also sizing design variables for one- and two-dimensional structural components, such as the thickness of plates. For such components, (2) is replaced by an equation relating stress resultants to generalized strains, and the sizing variables are included in $\mathbf{S}_c$.

Using (10)–(12), the equations of equilibrium, (3), can be differentiated to obtain the following continuum sensitivity equation:

$$\int_V \delta\epsilon_c : \mathbf{S}_c : \epsilon_c'\,\mathrm{d}V = \int_V \rho\mathbf{b}_c' \cdot \delta\mathbf{u}_c\,\mathrm{d}V + \int_A \mathbf{h}_c' \cdot \delta\mathbf{u}_c\,\mathrm{d}A - \int_V \delta\epsilon_c : \mathbf{S}_c' : \epsilon_c\,\mathrm{d}V \tag{13}$$

for all $\delta\mathbf{u}_c$ that belong to the space of kinematically admissible displacements. The left-hand side of (13) is the same as that of (3) if $\mathbf{u}_c$ is replaced by $\mathbf{u}_c'$. The right-hand side of (13) defines a pseudo-load (or fictitious-load), which explicitly depends on the design. Thus, solving the sensitivity equation, (13), is the same as solving the original structural equilibrium equation, (3), with different load terms. The major advantage of the continuum approach is that the sensitivity formulation is independent of discrete model and numerical schemes. The sensitivity equation is well defined as long as the functions on the right-hand side are integrable in the domain or on the boundary. Once the continuum sensitivity equation is obtained, it can be discretized in the same manner as the original analysis equations in order to obtain a system of matrix equations similar to (8). It is repeated that $\mathbf{u}_c'$ depends on the direction of the design change as contained in η.

When the design variables affect the shape of the domain, the differentiation of the equations of equilibrium is much more complicated because the integral domain depends on the design. Interested readers are referred to Refs. [6,7] for the material derivative approach, Refs. [45] or [46] for the control volume approach, and Ref. [47] for the Eulerian approach.

2.4. Discrete derivatives

In the previous section we have seen that the continuum sensitivity equations are derived by differentiating the governing continuum equations with respect to the design variables. Subsequently, a discretization takes place. For discrete derivatives this order is reversed, that is the discrete equation (8) is differentiated with respect to the design variables, giving

$$\mathbf{K}\mathbf{u}' = \mathbf{p}. \tag{14}$$

The pseudo-load vector $\mathbf{p}$ is defined by

$$\mathbf{p} = \mathbf{f}' - \mathbf{K}'\mathbf{u}. \tag{15}$$

It is clear from (14) that the design sensitivities require the solution of the same set of equations as solved for the response functions, but for another right-hand side (compare with (8)). The latter being the pseudo-load vector, see (15).

In calculating the pseudo-load vector, it is not necessary to differentiate the global load vector and stiffness matrix, but to differentiate only those elements that are affected by the design variable. The evaluation of the pseudo-load vector is then carried out by an assembly of all individual nodal points and finite element contributions. These contributions are obtained by differentiating the finite element stiffness matrices $\mathbf{K}_e$ with respect to the design variables and following a similar procedure for all load contributions.

The fact that the pseudo-load vector only depends on affected elements may be exploited to make the computation of the pseudo-load vector more efficient. For shape design variables this requires some additional attention. For that purpose one often tries to link the design variables only to boundary elements, which implies that only a boundary layer of elements is affected by the shape design variables.

The analytical differentiation process may become tedious. This especially holds true for shape design variables, although, symbolic computing software can help [48]. That is, symbolic computing software often features the automatic generation of the source code. Of course, this code must be integrated in the existing software. In any case, additional procedures must be implemented for each element used within the sensitivity analysis. The procedure must account for all possible design variables, and particularly for shape design variables as they are usually more complex than the original finite element routines. This type of discrete design sensitivities will be referred to as "analytical" discrete design sensitivities [49–51].

Because the pseudo-load vector generally requires significant implementation effort, approximations are frequently accepted for the pseudo-load vector that reduce this effort. These approximations particularly involve finite difference schemes for evaluation of the pseudo-load vector. Forward and central finite difference schemes are most popular. This type of design sensitivities is commonly denoted "semi-analytical" discrete design sensitivities.

Besides the options of symbolic computing and (semi-)analytical differentiation, there is also the option of applying computational or automated differentiation to a part of the analysis code. In this manner an elegant combination of analytical differentiation and automated differentiation can be achieved. The analytical steps, typically carried out at system level, ensure efficient use of computer resources, whereas the automated differentiation steps avoid expensive and tedious implementation. Automatic differentiation is typically applied to the element level [52]. A more extensive discussion on computational differentiation, focused on the differentiation of the entire analysis code, is provided in Section 2.5.

The discrete derivatives are the most commonly implemented, and they are available in several commercial finite element codes, for example, NASTRAN, GENESIS and MARC [53]. These implementations are typically based on analytical or semi-analytical formulations.

For shape design variables and analytical derivatives, we need to differentiate nodal locations with respect to design variables to obtain so called "design velocities". If available, these can also be exploited in a semi-analytical setting. However, for a semi-analytical formulation, the explicit construction of a design velocity field may be avoided by the introduction of design perturbations. Design perturbations also play a crucial role in the finite difference method. However, as discussed later, the design perturbations used have a pronounced effect on both consistency and efficiency.

For shape design variables, design perturbation involves both the size of the perturbation and its distribution over the domain. For the choice of perturbation size, considerations similar to those discussed for global finite differences are involved, see Section 3.2. Unfortunately, the semi-analytical formulation may be extremely sensitive with respect to this choice. We shall come back to this aspect extensively, and we only note here that this drawback may negate all advantages of a semi-analytical formulation and motivates modifications to the semi-analytical method. As demonstrated by Ref. [54], the size of the optimal design perturbation will be affected by the perturbation scheme applied.

2.5. Computational derivatives

Finally, computational, algorithmic or automatic differentiation refers to a differentiation of the computer code itself. Even if the finite element programs are composed of many complicated subroutines and functions, they are basically a collection of elementary functions. The computational (or automatic) differentiation method defines the partial derivatives of these elementary functions, and then the derivatives of complicated subroutines and functions are computed using propagation and the chain rule of differentiation. Thus, no approximations are introduced.

Without loss of generality, let us assume that an elementary function has two arguments, defined as

$$a = f_{\text{elem}}(z_i, z_j), \tag{16}$$

where $f_{\text{elem}}(\cdot,\cdot)$ represents $(++, \sin(\cdot), \ldots)$ operators for the single argument and $(+, -, *, /, \ldots)$ operators for the double arguments.

In the direct mode, the derivative of (16) can be defined as

$$\frac{\partial a}{\partial s} = \frac{\partial f_{\text{elem}}}{\partial z_i}\frac{\partial z_i}{\partial s} + \frac{\partial f_{\text{elem}}}{\partial z_j}\frac{\partial z_j}{\partial s}. \tag{17}$$

This derivative can propagate through complicated functions and subroutines using the chain rule of differentiation, which eventually produces the derivative of the structural response.

In the reverse mode, which corresponds to the adjoint mode in the previous sections, the derivatives are computed backwards through the computation. Due to the reverse procedure, this approach requires saving the entire function evaluation history, which, consequently, may require a significant amount of memory.

Computer programs that calculate the derivatives of output of other computer programs are now available and are applicable to ever-growing programs. The largest program that we found had about 800,000 lines [55]. Both first- and higher-order derivatives can be obtained [56,57]. Application of automatic differentiation to coupled systems is discussed by Ref. [58]. This approach was initially called automatic differentiation, but after a while it was realized that human intervention in the process is required in many cases in order to obtain a reasonably efficient code. So the name was generalized to computational differentiation.

As mentioned in the previous section, in order to achieve better performance automatic differentiation may only be used to parts of the source, this consequently leads to higher labor investment as compared to automatic differentiation of the entire source [56].

There are several automatic differentiation tools widely available today, notably ADIFOR (Automatic Differentiation of Fortran [59]) and ADOL-C for C/C++ programs [60]. In terms of implementation, there are two basic approaches to automatic differentiation—*source code transformation*, and *operator overloading*. Source code transformation can be viewed as a pre-compiler that adds code for computing the derivatives. Operator overloading is available in modern computer languages, such as C++ and Fortran 90, that provide the ability to redefine the meaning of elementary operators (such as multiplication) for various classes of variables. By defining new variable types that have gradient objects associated with them, and overloading the elementary operators to also produce gradients, the code can be transformed without increasing its size substantially. ADOL-C and ADOL-F [61] are examples of operator-overloading tools for automatic differentiation.

As pointed out by Ref. [62], the complex variable approach can also be viewed as a special case of computational differentiation. This approach was originally suggested by Refs. [63,64], and recently revived by Ref. [65]. It is based on the equation

$$f(s + \eta \mathrm{i}) = f(s) + \eta \mathrm{i} f'(s) - \frac{\eta^2 f''(s)}{2} - \frac{\eta^3 \mathrm{i} f'''(s)}{6} + \cdots, \tag{18}$$

which leads to the approximation

$$f'(s) \approx \mathrm{Im}\left[\frac{f(s+\mathrm{i}\eta)}{\eta}\right]. \tag{19}$$

That is, if we perform all the operations in complex arithmetic and replace a design variable s by $s + \mathrm{i}\eta$, with η being a small step size, we can estimate the derivative from (18). Improvement is achieved by the fact that subtraction of nearly equal numbers is omitted so that extremely small values of η can be used. Thus, the complex arithmetic provides the operator overloading needed for the differentiation, so that a standard compiler can do the work instead of specialized software.

The complex variable method has been applied in aerodynamic optimization and is found to be much less sensitive to step size than the standard implementation of forward differences [66,67]. However, application of the method might be hindered because complex arithmetic is not always available for all the functions and language constructs in the structural analysis software. In addition, the complex arithmetic may be more costly, with CPU time penalties ranging from a few percent to a factor of three [67]. In contrast to regular computational derivative approaches, the complex variable method does not feature an adjoint formulation or reverse mode.

3. Comparison for linear static problems

In the previous sections, an overview of different techniques for design sensitivity analysis has been given. A schematic overview of the techniques is provided in Fig. 2, where a rounded white box reflects an action, and a white rectangular box depicts an entity.

As mentioned earlier, the different options for sensitivity analysis may be subdivided into four classes: continuum or variational derivatives, global finite differences, discrete (semi-)analytical derivatives and computational or automated derivatives. In Fig. 2, these classes have been depicted using gray rounded boxes. This classification becomes somewhat blurred for the discrete derivatives as soon as either finite difference approximations or computational differentiation is blended in. Still we classify these hybrids as discrete (semi-/computational-)analytical derivatives because finite difference or computational differentiation for pseudo-loads still leads to the same adjoint calculations. In contrast, global finite differences do not have an adjoint option, while global computational differentiation has an adjoint method (reverse mode) that is substantially different from the discrete adjoint. The methods that rely on both analytical and computational differentiation still require manual implementation efforts.

Finally there is the question of name. Normally the discrete derivatives which rely on analytical derivatives and some finite difference approximations are denoted as "semi-analytical". The derivatives which combine analytical and computational differentiation could be referred to as "computational-analytical".

In the subsequent subsections, the aspects of accuracy and consistency, computational costs and implementation options and efforts will be highlighted (see Fig. 2).

3.1. Accuracy and consistency

When an exact method is used to find the perturbed solution, finite difference derivatives can be made to be accurate or consistent as is needed. For example, consider the problem of obtaining derivatives from a finite element model, created by a mesh generator with periodic adaptive meshing and error control. That is, for small design changes the mesh is perturbed by deforming it without changing the topology of the spatial discretization, while for larger changes the mesh may change by adding or removing elements. The numerical model is noisy, because as structural parameters vary, the finite element mesh will change in discrete steps to satisfy the error criterion [68]. Consistent derivatives of the numerical model can be

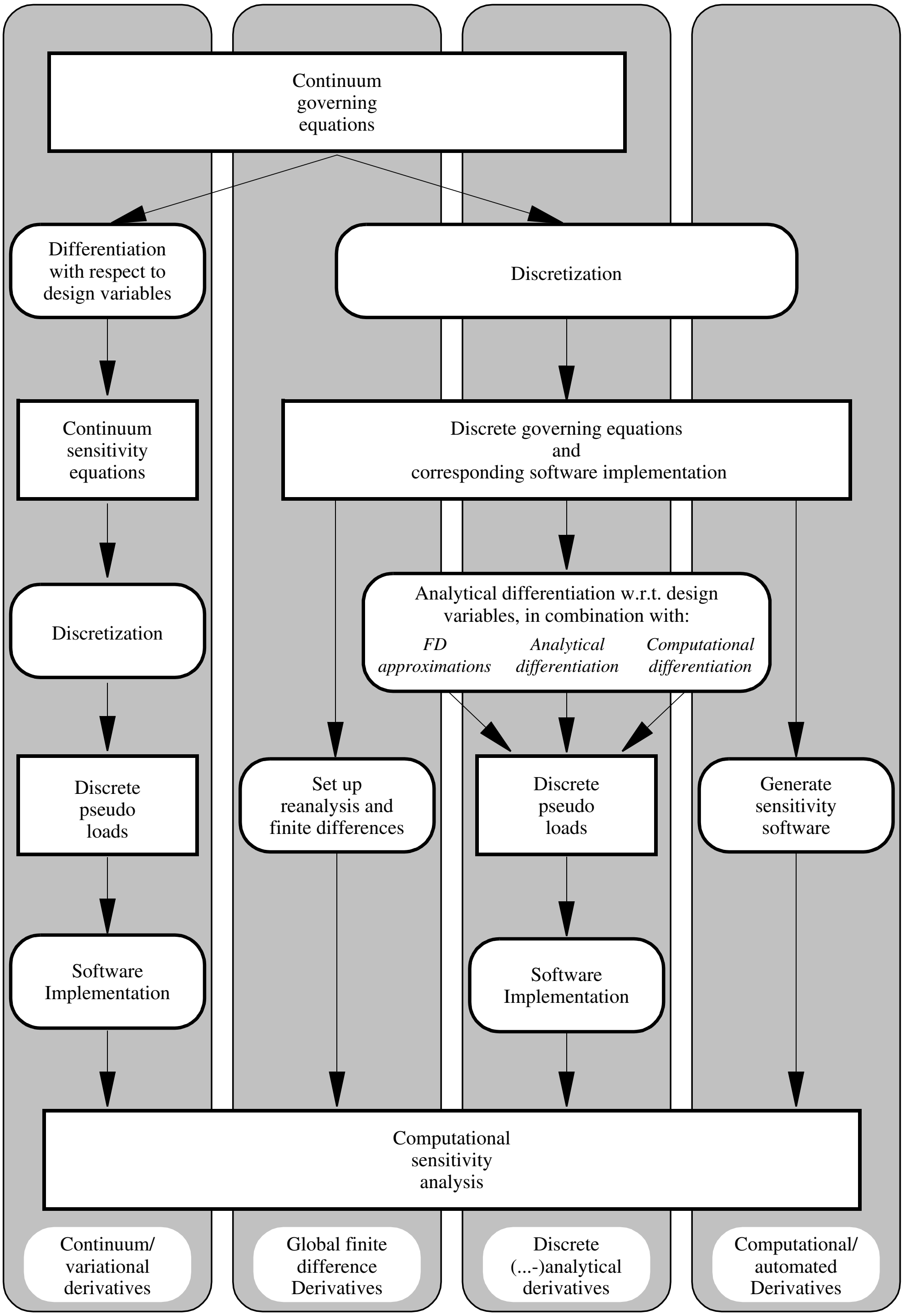
Continuum governing equations
Differentiation with respect to design variables
Discretization
Continuum sensitivity equations
Discrete governing equations and corresponding software implementation
Discretization
Analytical differentiation w.r.t. design variables, in combination with:
FD approximations
Analytical differentiation
Computational differentiation
Discrete pseudo loads
Set up reanalysis and finite differences
Discrete pseudo loads
Generate sensitivity software
Software Implementation
Software Implementation
Computational sensitivity analysis
Continuum/ variational derivatives
Global finite difference Derivatives
Discrete (...-)analytical derivatives
Computational/ automated Derivatives

Fig. 2. Overview of different techniques for design sensitivity analysis.

obtained by finite differences using steps that are small enough to avoid remeshing, or by not allowing remeshing with larger steps. If the error in the response itself is well conditioned, accurate derivatives of the exact response may be obtained by taking large finite difference steps and using high-order difference formulae in order to suppress the nonlinear effects associated with these large steps. The large steps clearly negate the effects of noise. However, when the discretization error changes with the design variables, then its derivatives determine the minimum achievable difference between computed and exact derivatives. Thus, to reach good accuracy discretization errors need to be kept constant (including their sign!), or, more realistically, sufficiently small by remeshing.

As noted in Section 2.4, the use of complex variables in a finite difference scheme can allow the use of extremely small step sizes without leading to condition noise errors [64,63,65]. The method resembles to a large extent a computational derivative scheme. Unlike regular computational derivatives, its use leads to approximations as compared to fully consistent derivatives. By taking a very small design perturbation, we see that truncation errors decrease quadratically. Consequently, the method will quickly converge to consistent derivatives. Similar to ordinary global finite difference schemes, the method will be sensitive to non-constant trends in the discretization errors. This implies that accurate derivatives can only be obtained provided the discretization error is well controlled. In the complex variable approach the need for higher-order schemes will be small as the effects of noise and truncation errors are minor.

In the context of accuracy and consistency there will be no principal difference between analytical, computational-analytical and computational derivatives. Therefore we shall refer to this group as discrete derivatives throughout the discussion on accuracy and consistency. Clearly this is not the case for the semi-analytical derivatives, which involve approximate steps within the differentiation process.

In terms of consistency, discrete derivatives can maintain a higher level of consistency than the continuum derivatives, even if the latter uses the same discretization with the original response. It has been demonstrated, that, in general, for the same discretization continuum and discrete methods may still be different [69,70]. The reason is because the former differentiates the continuum equations and then approximates using numerical techniques, while the latter differentiates the approximated discrete equations, see Fig. 2. Only when *all* steps involved are fully consistent with each other, can one expect identical results.

In the context of shape design variables it is important to realize that the mesh perturbation or the design velocity field should be consistent with the mesh updating scheme used for design updates. Reasons for having the mesh perturbation or the design velocity field inconsistent with the design updating are: (i) reduction of the error in semi-analytical sensitivities and (ii) efficiency considerations related to the evaluation of the pseudo-load vector. If such an inconsistency is introduced, then one must be aware that inconsistent derivatives may result. It must be stated however that standard semi-analytical design sensitivities may become less sensitive to the magnitude of the design perturbation by accepting such inconsistencies in the mesh perturbations. The consideration provided in the present paragraph is restricted to constant mesh topology. It becomes blurred as soon as design updates trigger modification of the mesh topology, e.g. to maintain modelling accuracy.

The accuracy of discrete and continuum derivatives is comparable. Like the global finite difference derivatives, the accuracy is to a large extent determined by the discretization error. This is, for example shown by Ref. [71], which provides a comparison of the accuracy of continuum, finite difference, and automatic differentiation for fluid flow with discontinuities due to shocks. This comparison shows that all three methods have similar accuracy properties.

The accuracy of discrete and continuum derivatives may be improved by using different spatial discretizations for the sensitivity and the response analyses. We need to set up mapping procedures and the factorization of the stiffness matrix may not be reused. The latter may be a severe penalty in terms of computational costs. Although the method may lead to better accuracy, inconsistencies will increase.

The accuracy of the semi-analytical results may be extremely sensitive to the choice of design perturbations, and may exhibit severe inaccuracies [72,73]. This is especially so for slender structures where the elements are subject to relatively large rotations. These accuracy problems have led to a series of studies that attempted to improve the accuracy of the semi-analytical shape design sensitivities. The proposed techniques can be subdivided into two groups. The first is related to improvements using higher-order or alternative finite difference schemes. The second group attempts improvements by more fundamental modifications at the level of single elements. Exponents of this group are Mlejnek's method [74], the "exact" formulation [75] and exact differentiation of rigid body modes [76,77]. Especially the latter two lead to rigorous accuracy improvement. The interested reader is referred to Appendix A for a more detailed discussion on recent progress in semi-analytical shape design sensitivities.

Recent work has demonstrated that the type of mesh perturbation may influence the accuracy of a semi-analytical formulation [78]. Different types of mesh perturbation were also studied by Ref. [54], who suggested the use of different values for boundary perturbations and interior perturbation.

Automatic differentiation is probably the most consistent means for obtaining derivatives because it actually follows computational operations step-by-step. When accuracy rather than consistency is desired, there are formulations that permit the use of automatic differentiation without corrupting the derivatives due to the effects of changing mesh. These formulations are based on application of the discretization to a reference domain of fixed shape instead of to the actual domain [79].

Finally, the chance of human errors either in the analytical derivation or computer implementation is important. Sandu et al. [80] make the point that automatic differentiation is the most error free-procedure for obtaining derivatives. However, this assumes that the automatic differentiation compiler is error free, which may not yet be true for presently available software. In addition, manual intervention is often desirable or needed in automatic differentiation, leading to the more general name "computational differentiation" [81].

3.2. Computational cost

The most commonly used global finite difference method employs forward or backward differences without any attempt to accelerate the re-analysis of the perturbed structure. This approach requires the cost of one analysis for each derivative. This cost is the reason for some of the other sensitivity calculation methods, but for a small to moderate number of design variables it is often quite affordable.

Methods that try to improve the accuracy of the finite difference approach, such as the use of higher order difference formulae and the complex variable approach increase the computational cost. On the other hand, methods that employ fast exact re-analysis, such as the Sherman–Morrison–Woodbury formulae [30], the iterative approach [31] and Kirsch's method [35], reduce that cost, but usually come at substantial implementation effort penalty. In the best case, these methods result in tools for design sensitivity analysis that are as efficient as discrete approaches.

In general, continuum derivatives and discrete derivatives are considered less expensive for the same level of accuracy than finite difference derivatives. The improvement in efficiency may be due to several factors: first, when the number of functions to be differentiated is much smaller than the number of design variables, an adjoint formulation can lead to substantial efficiencies. This advantage is also realized for automatic differentiation in which it is often called the reverse method [48]. Second, the sensitivity calculation does not require re-factoring of the stiffness matrix, and this benefits both a direct and an adjoint formulation.

The issue of computational cost between continuum and discrete methods appears to be problem- and implementation-dependent. Burczynski et al. [82] provide a comparison of cost and implementation issues for the continuum and discrete methods for boundary element formulations. Unfortunately, their comparison assumes that the continuum method is implemented with an adjoint approach while the discrete formulation with the direct approach. So the comparison reflects the advantages of the adjoint over direct approaches as much as those of the continuum over discrete methods. In general, if the continuum and discrete methods are implemented within the finite element source programs, the difference in computational cost between the two methods would be negligible. If the continuum method is implemented out of the finite element programs, then its efficiency may be less than that of the discrete method.

For shape sensitivity problems, boundary methods are available which reduce the computational effort associated with the calculation of the pseudo loads. This holds true for continuum sensitivities [83,84] as well as for discrete derivatives.

The computational cost of finite element based discrete design sensitivities can be reduced by introducing mesh perturbations that affect only boundary elements. Then the calculation of the pseudo-load vector only involves a limited number of elements [85]. However, in the optimization procedure, this method can yield undesirable mesh geometry. Moreover, if the mesh perturbations and the design updates are inconsistent, then we must expect a penalty in terms of inconsistency of the design sensitivities.

In boundary element approaches, Erman and Fenner [86] proposed a method that allows design points to move in one direction, which is normal to the boundary surface. They report a factor of three savings in computer time.

So far we have assumed that a direct type solver has been applied. When an iterative matrix solver is employed, the cost of sensitivity calculation can be higher than the global finite difference derivatives because the former needs to construct pseudo-load vector in addition to solving the sensitivity matrix equation. If the iterative solver is augmented using preconditioning techniques, then computer time savings may result from the fact that investments made for preconditioning can be reused.

The efficiency of computational differentiation appears to be poor for the direct method, and it appears to vary widely depending on the computer used. Most of the results come from CFD applications. For example, Carle et al. [87] compare the calculation of derivatives of the lift to drag ratio of a wing with respect to 88 geometry design variables. The direct (forward) approach required computation times of 300–700 (depending on the computer used) times the cost of function evaluations, compared to 89 for finite differences. The adjoint (iterated reverse) method required computation times of only 7–21 times the cost of function evaluation. For helicopter structural vibration problems, Walsh et al. [88] cite a ratio of about 1.75 between automatic differentiation and finite differences. Hu [89] similarly finds a ratio of 1.9 for CFD rotor calculations. In addition to the increased CPU requirements, automatic differentiation typically requires large increases in memory, especially the adjoint version.

3.3. Implementation options and effort

The global finite differences method is usually the easiest to implement, and it is probably the most widely used method for obtaining derivatives. Obviously, it can be used entirely without access to the source code. When it is used on the basis of a perturbation from a specific configuration, i.e. when the analysis of a perturbed design is not started from scratch but is initiated with the solution corresponding to the nominal design, then limited access to the source code may be required.

On the other hand, as noted in Section 3.2, attempts to reduce the cost of the method can come at a substantial implementation effort. The Sherman–Morrison–Woodbury formulae [30], for example, can be implemented in many structural analysis programs without access to the source code, but with substantial effort to create the right-hand sides from the perturbation in **K**.

Most structural analysis programs that offer structural sensitivity calculations use the discrete method implemented with the semi-analytical approximation. This has the advantage that it requires very little programming effort and almost no element-dependent sensitivity routines. The analytical discrete derivatives have the advantage of introducing no approximations, but the associated implementation efforts can be tremendous. An alternative is to carry out the differentiation with either symbolic computing software and to generate the associate source code automatically or to use computational differentiation on a part of the code. In this case, computational differentiation or symbolic computing is typically applied to the element routines only. Some manual implementation is still needed, although substantial savings can be achieved.

For a user with a "black box" program, without access to the source code, implementation the discrete or continuum derivatives depends on the availability of access to the program data base, and on the possibility of adding user-defined subroutines. Poldneff et al. [53] provide a discussion on the pros and cons for implementing design sensitivities either inside or outside a finite element program. Finite element software often provides users with a programming language (e.g. DMAP in NASTRAN) that allows manipulation of stiffness matrices, stresses and displacements without access to the source code. These can facilitate implementation of the discrete or continuum sensitivities. Akgün et al. [90] describe the use of such a language with the EAL software for implementing continuum derivatives.

Choi and Duan [41] contend that implementing the continuum approach may be less dependent on the availability of the source code than the discrete approach. They implemented their continuum derivatives outside the finite element program ABAQUS. Using the displacement data from the structural response, the pseudo-load vector is constructed out of the finite element program. After that, the displacement sensitivity was calculated using the restart capability of ABAQUS with the pseudo-load vector. A similar approach was used by Ref. [91] with the EAL program. The separation of the sensitivity program from the finite element program may also provide more flexibility in implementing options. For example, Chang et al. [92] implemented the sensitivity calculation module that can connect with such finite element programs as ANSYS, ABAQUS, and NASTRAN.

Computational differentiation holds the promise of producing high accuracy with little implementation effort. However, the associated compilers have not matured yet to the point where this is generally true. Therefore, depending on the details of the original code, the use of software such as ADIFOR or ADOL-C can require almost no effort or a great deal of effort. In general the longer the program to be differentiated, the higher the chance of implementation problems.

4. Transient analyses

4.1. Introduction

In the present section, design sensitivity analysis for linear transient response will be reviewed. In contrast to the preceding sections, this discussion will be mainly focused on global finite differences and discrete derivatives. The reason is that the focus is on temporal discretization and not on spatial discretization. For the latter the reader is referred back to the previous sections. In the temporal domain, the majority of the papers starts from a time discretization, rather than a continuous time description, or addresses the use of global finite differences.

When a time dependent load and/or boundary condition is applied to a structure, the transient response of the structure is important. The transient dynamic problem of a structure is often called the initial-boundary-value problem (IBVP). In such a case, *Lagrange's equation of motion* becomes a second-order differential equation, as

$$\mathbf{Ma}(t) + \mathbf{Cv}(t) + \mathbf{Ku}(t) = \mathbf{p}(t), \tag{20}$$

with the following initial conditions:

$$\begin{cases} \mathbf{u}(0) = \mathbf{u}_0, \\ \mathbf{v}(0) = \mathbf{v}_0, \end{cases} \tag{21}$$

where $\mathbf{a}(t)$ and $\mathbf{v}(t)$ are the acceleration and velocity vectors, respectively.

The IBVP in (20) and (21) must be satisfied for all time period $t \in [0, T]$. In practice, however, (20) is imposed in discrete time intervals, called *temporal discretization*. Using response results at the previous time, the response at the current time is calculated using the time integration method. Methods of integration for the equation of motion in a dynamic response analysis can be implicit and explicit.

Among many methods, the Newmark family of time integration methods will be discussed, in which a predictor–corrector scheme is often employed. Let subscripts n and $n + 1$ represent the time steps t_n and t_{n+1}, respectively. When dynamic responses at time t_n are available, the velocity and displacement at time t_{n+1} can be integrated by

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_{\mathrm{pr}} + \gamma \Delta t \mathbf{a}_{n+1}, \\ \mathbf{u}_{n+1} = \mathbf{u}_{\mathrm{pr}} + \beta \Delta t^2 \mathbf{a}_{n+1}, \end{cases} \tag{22}$$

where $\mathbf{v}_{\mathrm{pr}} = \mathbf{v}_n + (1 - \gamma)\Delta t \mathbf{a}_n$ is the velocity predictor and $\mathbf{u}_{\mathrm{pr}} = \mathbf{u}_n + \Delta t \mathbf{v}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2 \mathbf{a}_n$ is the displacement predictor. In (22), β and γ are Newmark integration parameters.

Using (22), (20) at time t_{n+1} can be expressed in terms of the acceleration vector, as

$$\left[\mathbf{M} + \gamma \Delta t \mathbf{C} + \beta \Delta t^2 \mathbf{K}\right] \mathbf{a}_{n+1} = \mathbf{p}_{n+1} - \mathbf{K}\mathbf{u}_{\mathrm{pr}} - \mathbf{C}\mathbf{v}_{\mathrm{pr}}, \tag{23}$$

which solves for $\mathbf{a}_{n+1}$. After that, the velocity and acceleration vectors are corrected using (22).

The stability and accuracy of the time integration method for a linear system is thoroughly examined by Ref. [93]. The unconditionally stable condition for the Newmark family integration method is given by $2\beta \geqslant \gamma \geqslant \frac{1}{2}$, and second-order accuracy is preserved only when $\gamma = \frac{1}{2}$, which does not show any numerical damping effects. Choosing a different value for γ ($>\frac{1}{2}$) shows first-order accuracy with numerical damping effects. When $\beta \neq 0$, (22) is implicit because the displacement and velocity at t_{n+1} are functions of the acceleration at t_{n+1}.

The explicit integration method corresponds to the case in which $\beta = 0$ and $\gamma = \frac{1}{2}$ with diagonal matrices of $\mathbf{K}$ and $\mathbf{C}$. In conjunction with the lumped mass matrix, solution of (23) becomes very efficient because (20) becomes a set of linear algebraic equations. However, this scheme is only conditionally stable so that very small time-step sizes are required to achieve numerical stability. The time step size is governed by the length of the smallest element and by the material properties. Mathematically, the time step size is determined so that the next time step can be within the domain of influence in the hyperbolic system. Another explicit time integration method that is popular in applications is to use an intermediate time-step to integrate the velocity vector, i.e., at $t_{n+1/2}$.

For design sensitivity purposes, the following two approaches yield the same results: (i) differentiating (20) and introducing temporal discretization, or (ii) differentiating (23). By assuming the time interval is independent of design s, differentiation of (23) yields the following sensitivity equation:

$$\left[\mathbf{M} + \gamma \Delta t \mathbf{C} + \beta \Delta t^2 \mathbf{K}\right] \mathbf{a}'_{n+1} = \mathbf{p}'_{n+1} - \mathbf{K}\mathbf{u}'_{\mathrm{pr}} - \mathbf{K}'\mathbf{u}_{n+1} - \mathbf{C}\mathbf{v}'_{\mathrm{pr}} - \mathbf{C}'\mathbf{v}_{n+1} - \mathbf{M}'\mathbf{a}_{n+1}. \tag{24}$$

Clearly, this set of equations must be accompanied with a proper set of starting conditions. After solving for the acceleration derivative, the derivatives of displacement and velocity are updated in a manner similar to (22). In addition to the derivatives of stiffness and mass matrices, (24) requires the derivative of displacement at the previous time step, which makes the sensitivity equation history dependent. In the implicit method, the factored coefficient matrix from the response analysis can be reused in solving the sensitivity equation efficiently. In the explicit method, the sensitivity analysis becomes more expensive than the response analysis because calculating the right-hand side of (24) is more computationally expensive than that of (23) and the sensitivity analysis cannot take advantage of the factored coefficient matrix from the response analysis.

It is well-known that the adjoint variable method for transient dynamic problems with an initial condition becomes a terminal value problem for which a terminal condition is given for an adjoint equation [94]. Thus, the adjoint equation cannot be solved simultaneously with the response analysis. This fact significantly complicates calculations associated with transient dynamic design sensitivity analysis using the adjoint variable method.

Linear structural dynamics is commonly solved using a reduced basis approach, with the reduced basis consisting of vibration modes and additional base vectors, often called Ritz vectors. That is, we write the displacement as

$$\mathbf{u} = \mathbf{\Phi}\mathbf{q}, \tag{25}$$

where $\mathbf{\Phi}$ is a matrix composed of the base vectors. The equations of motion are then reduced to

$$\mathbf{M}_R\mathbf{a}_q(t) + \mathbf{C}_R\mathbf{v}_q(t) + \mathbf{K}_R\mathbf{q} = \mathbf{p}_R, \tag{26}$$

where

$$\mathbf{M}_R = \mathbf{\Phi}^{\mathrm{T}}\mathbf{M}\mathbf{\Phi}, \quad \mathbf{C}_R = \mathbf{\Phi}^{\mathrm{T}}\mathbf{C}\mathbf{\Phi}, \quad \mathbf{K}_R = \mathbf{\Phi}^{\mathrm{T}}\mathbf{K}\mathbf{\Phi}, \quad \mathbf{p}_R = \mathbf{\Phi}^{\mathrm{T}}\mathbf{p}. \tag{27}$$

For most problems, sufficient accuracy can be obtained with a number of basis vectors in $\mathbf{\Phi}$, which is a small fraction of the number of degrees of freedom in $\mathbf{u}$. Thus, the modal reduction can greatly lower the computational cost of the transient analysis. When the natural vibration modes are used as basis vectors the reduced mass and stiffness matrices are diagonal, and the reduced damping matrix can also be diagonal for proportional damping. This allows uncoupled solution of the reduced equations of motion. For this case, there is also a correction to account for the neglected modes, called the mode acceleration method.

The differentiation of (26) can proceed as for the original equations, (20). However, there is the issue of whether to differentiate the basis vectors or treat them as constant. Greene and Haftka [95] provide some discussion on the accuracy and cost tradeoffs associated with this choice. They also provide a derivation and results of numerical experiments with the mode acceleration method.

With the modal approach, it is possible that the order of the reduced system is smaller than both the number of design variables and the number of desired response quantities. In that case, the Green Function method [96,97] may be superior to both the direct and adjoint methods.

4.2. Accuracy and consistency

For transient analyses, using explicit integration, it may be necessary to apply relatively tight error tolerances in order to achieve sufficiently accurate design sensitivities [98].

When the time step is automatically determined based on the analysis response at the current time, the time step also depends on the design variable [99]. The accuracy of the sensitivity results drops dramatically when such a variable time step effect is not considered in sensitivity calculation. Cho and Choi [99] use a fixed time step claiming that the accuracy of response analysis is assured when the time step is small enough. In the implicit method, such a problem does not occur since the sensitivity equation is solved after the response analysis is converged [100]. These problems for explicit integration improve the attractiveness of finite difference derivatives for explicit integration. Haftka and Malkus [101] provide estimates of the optimal time step.

4.3. Computational cost

Special attention must be paid to transient analysis. For this type of analysis the cost for the derivative calculation will greatly depend on how many of the (intermediate) results of the response evaluation can be reused for economizing the sensitivity calculation. This turns out to be mainly determined by the choice of the integration scheme used, i.e. whether an explicit or an implicit integration scheme will be used. Work using explicit integration and direct differentiation is, e.g., reported in Refs. [53,102,103], whereas the adjoint method is described in Ref. [103]. Stillman [50] gives some estimates on this aspect in the context of an explicit time integration scheme applied to problems including plasticity and contact. Stillman [50] observes a factor two computer time savings for an analytical implementation as compared to global finite differences. These savings are mainly due to contact becoming easier for sensitivity analysis. Karaoglan and Noor [104] show similar findings using implicit integration schemes. Moreover it is noted that the design sensitivity analysis for this type of transient problems is more suited for parallel computing than the response evaluation itself. This is due to the fact that the sensitivity analysis is more apt for scaling. The observations of Ref. [50] are consistent with other observations. Cho and Choi [99] sketch an even more pessimistic picture, namely for explicit time integration the computing time for the sensitivity analysis may be even longer than the time required for the response analysis. This would basically imply that under such circumstances there is no need to implement sensitivity analysis other than on the basis of global finite differences. A similar conclusion was reached by Ref. [105] for sensitivities of transient thermal response. It is also possible to parallelize the calculation of the nominal and perturbed solutions needed for the global finite difference approach [101].

For implicit transient analyses using a direct method the picture is entirely different. For implicit analyses a factorized tangent matrix can be reused for the sensitivity analysis. Moreover, whereas the response function may need an incremental-iterative type of calculation, the sensitivities only call for an incremental update. Typical examples can be found in Refs. [106,104,98,39,107]. For a car bumper design problem [39] show that the sensitivity analysis for a single design variable requires only a few percent of the response evaluation. The calculation of the required pseudo loads can be either carried out analytically [39] or by means of semi-analytical approximations [107]. In many cases an adjoint formulation is more effective in terms of computer time than a direct one. Recent progress in adjoint sensitivities for implicit time integration schemes can be found in Refs. [43,106]. A disadvantage of transient analyses is the required backward integration [94]. As a consequence, significantly more memory storage is required, which may influence the effectiveness of the adjoint method for transient analyses [43,106]. Note that for a direct method the sensitivities are obtained simultaneously with the response evaluation, i.e. using a forward integration.

Besides the difficulty associated with the need to integrate from the final time to the initial time, the adjoint method suffers from another computational cost advantage for transient problems: we tend to have more response quantities of interest than in static problems. For example, a displacement constraint for static analysis can often be localized to a single point in the body where the displacement is maximal (like the tip of a cantilever beam). However, in transient response we need to enforce the constraint over the entire time span, and it may not be possible to predict ahead of time when the displacement is maximal. Therefore, it is common to check the constraint at a grid of time points, which converts it to a large number of constraints, thus handicapping the adjoint method. To alleviate this problem, it is possible to follow only the peaks of the response, since the first derivative of a smooth maximum can be calculated by assuming that the maximum is stationary [97].

4.4. Implementation options and effort

The implementation effort of transient dynamic problems may not be increased significantly compared to static problems. When damping contribution is ignored, the only additional term is the inertia, which is simpler than the stiffness. In fact, many finite element programs use a single code to handle static and dynamic problems together. The implementation effort related to the derivative of the stiffness matrix will be the same as for static problems.

An implementation of a direct method for transient analysis largely depends on the approach followed. As indicated by Ref. [50] an analytical implementation of the pseudo loads may be tedious in the sense that most of the component source code needs modification. However, application of finite difference approximations for the calculation of the pseudo-loads may alleviate the implementation efforts considerably [98,107]. Implementation of an adjoint approach is significantly complicated due to the fact that the adjoint equations cannot be solved simultaneously with the response evaluation but require a backward integration [98]. Nguyen et al. [108] state that because a direct method is always carried out simultaneously with the response evaluation it is less flexible. Since a backward integration is required for an adjoint sensitivity analysis, it is carried out after the response evaluation, which may give additional flexibility to the user.

Implementing a design sensitivity program (other than finite differences) outside of the response evaluation program is impractical due to the significant amount of data required to be stored. In the implicit method, the factorized coefficient matrices as well as response results (displacements, velocities, and accelerations) must be stored at each discrete time step. Thus, the availability of the source code can be a critical factor in sensitivity analysis of transient problems.

5. Conclusions and discussion

Four different options for calculating sensitivity information in the context of accuracy and consistency, computational cost, and implementation options and effort are summarized. The global finite difference method is the most convenient in implementation, but high cost and difficulty in finding appropriate perturbation size are disadvantages. The continuum method has advantages in theoretical soundness, low cost, consistency, and possible different meshes for response and sensitivity. However, it requires more mathematical understanding. The discrete method has advantages in low cost and consistency, but has disadvantages in the requirement of the source code and dependence on perturbation size for the semi-analytical method. The latter complication may become extremely pronounced for a "standard" semi-analytical implementation when applied to shape design sensitivities. However, this complication can be tackled by using the "refined" or "exact" semi-analytical methods. The computational derivative is the most consistent among four methods. However, computational cost is usually higher than other methods and practical for small sized programs. Application of computational differentiation to parts of the source code seems a very promising route. It may be a very good compromise between computational efficiency and implementation effort, since it benefits from a partial analytical formulation, whereas the tedious implementation is left for the automated differentiation process. Moreover, the differentiation process does not involve any approximation, as is the case for the semi-analytical variant.

When human time is of the essence and there is little information about the analysis code, then global finite difference is the method of choice. When the analysis code is small, the sensitivity analysis is going to be used for a long time in the future, but computational costs are not overwhelming, then automatic differentiation may be the way to go. When there is a black box code, with no access to stiffness matrices, global finite differences appear to suffer from accuracy, there are many design variables, continuum derivatives may be called for. Finally, for general software developers, the discrete method with semi-analytical method appears to be the method of choice. If accuracy is problematic, then the use of improved semi-analytical methods can be considered or analytical methods. The latter may be labor intensive and the use of symbolic differentiation or automatic differentiation of parts of the software is highly recommended.

In transient problems, an elaborate sensitivity calculation only benefits from the implicit time integration method. In the case of the explicit method, the global finite difference method is generally the most efficient option. In some situations, however, it is almost impossible to find an appropriate perturbation size for the finite difference method. In such a case, the usage of other sensitivity calculation methods is justified.

Acknowledgments

The first author greatly acknowledges the support by the Koiter Institute Delft, which enabled him to initiate a collaboration with the University of Florida, Gainesville. The authors also acknowledge the support by Xueyong Qu and Matthijs Langelaar.

Appendix A. Semi-analytical method

A.1. Introduction

The term "semi-analytical" has been used to describe different procedures in different disciplines. In the area of structural sensitivity calculation, the term semi-analytical design sensitivities is generally connected to analytical differentiation of the discrete governing equations, whereas the derivatives of their coefficients are calculated on the basis of finite differences. Often this implies that finite differences are applied at the element level, whereas the analytical differentiation applies to system level, see, e.g., Ref. [1] and the references therein. A description for transient analyses is provided by Ref. [107]. For limit points the reader is referred to Refs. [109,110,36]. Linear eigenvalue problems are addressed in Refs. [97,111]. For multidisciplinary examples the reader is referred to Ref. [112].

Analytical calculation of the pseudo-load vector **p** in (15) is fairly simple for certain design variables. A typical example is the thickness of a homogeneous plate or shell. Here the calculation of the pseudo-load vector can easily be realized analytically, e.g., using commercial finite element software and user-defined subroutines, see Ref. [113]. Treatment of more complex design variables, for example, shape design variables, becomes more involved. In a semi-analytical formulation the enormous effort required to implement analytical derivatives of **f** and **K** is circumvented by using finite differences approximations. At this stage, the differences between analytical and semi-analytical discrete derivatives become evident. The perturbed system stiffness matrix can be created and the derivatives approximated by subtracting the perturbed and original matrix and dividing by the perturbation step size. However, often it is more efficient to assemble the pseudo-load vector (**p**) from individual elements and nodal point contributions. This is the case when the design variable affects only a small portion of the stiffness matrix, or when a stiffness matrix requires significant memory allocation. Normally, no new routines are required for this assembly process as such procedures would be available in an existing finite element code.

A.2. Accuracy and consistency

Provided sufficiently small perturbations are used and round-off errors remain sufficiently small, semi-analytical derivatives are consistent. Although good results may be found using semi-analytical derivatives [114,43], severe accuracy problems, i.e. in the sense of inconsistencies, have been observed for shape design variables as well as for stiffness design variables in nearly incompressible materials [72]. These accuracy problems may negate all advantages of a semi-analytical formulation. To be more precise, the resulting accuracy may become extremely sensitive to the selection of the perturbation parameter used in finite difference approximations at element level. In those cases, the range of applicable design perturbations may become very small or non-existent. These accuracy problems have been observed by Ref. [73], and studied in Refs. [72,115–118], among others. It has been found that complications are related to rigid body rotations of individual finite elements [73,119,120].

To enlarge the range of applicable design perturbations, the use of higher-order finite difference schemes has been proposed [72,115,118]. Somewhat similar is the simultaneous use of both forward and backward finite difference schemes [115] or the use of a second-order correction [121]. Typical examples of higher-order finite differences can be found in Refs. [53,98]. Higher-order finite difference schemes are easy to implement in an existing semi-analytical code and alleviate the accuracy problem. However, there are two drawbacks. Firstly, more design perturbations are called for, which implies degradation of computational efficiency. This is caused by additional effort for both mesh regeneration and calculation of the corresponding contribution to the pseudo-load vector. Secondly, using a simple beam example [73,72,116,118,122], it can be shown that the accuracy problem is not solved rigorously by means of a higher-order finite difference scheme. It does improve the accuracy substantially.

Inspired by the fact that in shape sensitivity severe inaccuracies are related to rigid body modes, Mlejnek [74] proposed a modified finite difference scheme that conserves the consistency conditions for rigid body modes and their design sensitivities. Implementation of the method is easy, provided that the finite element code at hand is based on the so-called "natural approach" [123]. Its implementation in a more general finite element code becomes more involved. Computational efficiency of the method is nearly as good as a standard formulation, as only relatively simple additional operations at element level are to be carried out. The method has only been described for linear static analysis.

Based on specific features of the finite element matrices, Olhoff et al. [75] constructed the so-called "exact" semi-analytical formulation. The method leads to exact derivatives. Application to linear static analysis has been addressed by Refs. [75,124]. Eigenvalue sensitivities have been addressed by Ref. [111]. Like Mlejnek's formulation, the "exact" formulation also affects element level only. However, information on the structure of the elements has to be retrieved, which might be a disadvantage and causes the "exact" formulation to become somewhat more element-dependent. The big advantage of the "exact" formulation is that it eliminates the accuracy problem rigorously. To the best of the authors' knowledge, it has only been demonstrated for linear static analysis and eigenvalue problems. A disadvantage seems to be that it may be somewhat difficult to implement for complex elements. For this reason Hinton et al. [124] have used a simplification of the "exact" method for the Huang-Hinton shell element.

A refined semi-analytical formulation based on exact derivatives of rigid body modes corresponding to individual elements has been. For linear static analysis a description is given in Ref. [76], whereas linearized buckling is addressed in Ref. [125] and geometrically nonlinear and limit points are dealt with in Ref. [77]. There seems to be no principal reason that would prevent application of the same idea to fully nonlinear problems. Two key concepts are explored. First, in the case of linear static analysis and eigenvalue problems, the nodal degrees of freedom are decomposed at element level into a deformational component and rigid body modes. Second, the pseudo-load vectors are decomposed at element level into self-equilibrating components and non-self-equilibrating ones. Implementation of this refined semi-analytical formulation requires specification of the rigid body modes and their design sensitivities for the individual elements. These rigid body modes can be specified as soon as knowledge on the nodal degrees of freedom is available, i.e. *no details on the element formulation are required*. As a consequence, the elements can be entirely looked upon as black boxes. The investment for implementation is limited and can be carried out to a large extent generically. The only element-dependent component is the definition of the rigid body modes, which relate to the nodal degrees of freedom used. The refined semi-analytical formulation conserves the computational efficiency of a semi-analytical formulation and eliminates accuracy problems rigorously. Like Mlejnek's method and the "exact" formulation, calculation of the pseudo-load vector becomes a little more expensive. In all cases it still scales linearly with the number of elements subject to perturbation. The additional computational effort is comparable with Mlejnek's formulation and the "exact" formulation. Finally it is noted that the refined formulation can be combined with any (higher-order) finite difference scheme. Although not supported yet by rigorous testing, it seems that the differentiation of the rigid body modes may be carried out by finite differences.

The refined semi-analytical concept has also been studied in the context of second-order semi-analytical design sensitivities in Ref. [126].

Parente and Vaz [127] proposed a refined semi-analytical method that can accurately evaluate the contributions from the rigid-body motions. The derivative of the internal force term is decomposed into components that belong to the space that is spanned by rigid-body modes and those that belong to its orthogonal subspace. While the latter can be accurately evaluated using finite difference method, the former causes problems when rigid-body rotations are present. From a free-body equilibrium condition, the former is represented by the derivatives of the rigid-body modes, which can be accurately evaluated.

In the context of higher-order semi-analytical derivatives, the work of Bernard et al. [128] is relevant. They calculate higher-order derivatives, i.e. third order and higher, for mass- and stiffness matrices using interpolations. There seems, however, no argument to support that this interpolation based approach does not lead to inaccuracy problems similar to those observed for standard semi-analytical approaches.

Semi-analytical derivatives for hybrid finite elements have been discussed by Ref. [129]. They observe that the semi-analytical derivatives perform better for mixed elements as compared to standard displacement based elements. In our opinion this observation should be attributed to the fact that element defects have been removed by the hybrid formulation, rather than a rigorous fundamental improvement of the semi-analytical formulation. As a matter of fact, the simple beam example which has been used in many publications to demonstrate and study the defects in semi-analytical derivatives could be formulated using a hybrid formulation.

A.3. Computational cost

Assuming that mesh perturbations are generated efficiently, the semi-analytical formulation is efficient in terms of computing costs. The effort for calculating the pseudo-load vector is proportional to the number of elements subject to perturbation and is therefore in the worst case comparable with the assembly of the **K**-matrix. The higher-order finite difference schemes used to improve accuracy affect the computational cost. Additional costs for the ‘‘exact’’ or the ‘‘refined’’ formulations are minor and scale linearly with the number of elements.

There have also been attempts to improve mesh perturbations such that the severe inaccuracies become milder. Unfortunately, these methods may have a large influence on the computational efficiency of the mesh adaptation algorithms, moreover their effect on accuracy is not rigorous [120,130].

Considerations on how the resulting sets of equations are to be solved do not differ from those discussed previously for discrete sensitivities in general.

A.4. Implementation options and effort

Implementation of the semi-analytical method is generally simple and straightforward. First of all, a method for the generation of a perturbed discretized model must be implemented. A point of attention might be that local perturbations may become too large or too small in shape optimization. This particularly happens when a model includes local mesh refinements. This complication can be avoided if the perturbations are selected for individual elements or groups of elements. Unfortunately, implementation of such a more advanced perturbation scheme becomes involved. Perturbed meshes should be generated in a cost effective way. Remeshing from scratch is attractive from the point of implementation, however, it is generally too expensive computationally. Another complication is that mesh topology might be affected, which would obstruct use of the finite difference scheme. As a consequence, modifications in the mesh generator, especially if curved surfaces are being used, will be generally unavoidable.

The next step is implementation of the finite difference scheme to calculate the pseudo-load vector within the finite element code at hand. Here no knowledge on the specific structure of **f** and **K** is required, i.e. there is no need for details on the specific element formulation used. Hence, available routines for elements and nodes can be dealt with as black boxes. Provided that routines for calculating element and nodal contributions to **K** and **f** are available, calculation of the pseudo-load vector can be set up generically. To determine the design sensitivities of **u**, a set of equations similar to the one solved for calculation of **u** must be solved. Thus, the same solvers as used for a finite element analysis can be used for calculating design sensitivities. Coding of derivatives of generalized stress and deformations, can again be done generically. The coding of the derivatives of stress and strain components at the integration points is generally more involved and calls for information on the specific finite element definition. The same holds true for stress and strain criteria. Note, that these observations also apply to a certain extent for analytical formulations. In conclusion, limited access to the source code is required and even preferred, see Ref. [53]. Generally, it will be relatively easy to implement semi-analytical design sensitivities, using, for example, user-defined subroutines. The required modifications in the mesh generator at hand will be more tedious.

Particularly within a linear setting, different options are available for implementing semi-analytical derivatives. To be more precise, the way the pseudo-load vector is calculated offers mainly two alternatives (see Ref. [131]). Which of these alternatives is most efficient depends on the nature of the design velocity field.

Appendix B. Indices to references included

B.1. Entries based on type of derivatives

Global finite differences: [33,34,132–134].
Iterative global finite differences: [31,135].
Analytical discrete derivatives: [10,11,49–51,69,102,104,107,108,114,132,133,136].
Semi-analytical discrete derivatives: [33,34,54,72–76,98,106,111,112,115–119,121,122,124–127,129,131,135,137–142].
Computational derivatives: [48,57,58,79,81,143–147].
Continuum derivatives: [12–14,17,37,38,40,41,44,45,69,82,83,86,148–156].
Higher-order derivatives: [13,40,57,98,126,128,136,148,150,157–159].

B.2. Entries based on type of modeling and analysis

Adaptive modeling: [160–162].
Aerodynamics: [4,56,83,146].
Aeroelastic: [9,11,19,112].
Boundary elements: [37,38,40,42,82,86,133,135].
Contact: [39,50,102,104,163–167].
Convection: [141,153,154,162].
Coupled: [9,58,138,140].
Dynamics: [39,43,50,99,100,104,106,107,168–172].
Eigenvalue: [31,77,111,125,128,158,159,173–201].
Fluids: [21,66,68,71,87,89,141,145,153,154,156,160,161].
Fluid-structure interaction: [17,140].
Fracture: [12–14].
Hyperelasticity: [41,149,163,166,201].
Kinematics: [152].
Limit points: [36,51,203–208].
Linear statics: [31,33,34,37,38,42,44,45,49,54,57,69,72–76,86,114–117,119,121,122,124,126,129,131,133,135,143,148,150].
Meshfree methods: [39,100,163–167,202,209–213].
Multi-body systems: [81,107].
Path-independent statics: [36,51,77,127,137,203,214].
Plasticity: [39,50,99,100,106,139,151,155,164,167,209,211,215,216].
Post-buckling: [136,217,218].
Softening: [214].
Vibration: [17,219].
Viscoelasticity: [98].

B.3. Methods for design velocity field calculation

Finite difference method: [85,220–222].
Iso-parametric mapping method: [223–230].
Boundary displacement method: [227–229,231–234].
Hybrid method: [235,236].
Physical approach: [91,237].

B.4. Related review papers

Refs. [1–5,238,239].

References

[1] R.T. Haftka, H.M. Adelman, Recent developments in structural sensitivity analysis, Struct. Optim. 1 (1989) 137–151.

[2] B.M. Kwak, A review on shape optimal design and sensitivity analysis, Struct. Engrg./Earthq. Engrg. 10 (4) (1994) 159s–174s.

[3] Y.L. Hsu, A review of structural shape optimization, Comput. Ind. 26 (1994) 3–13.

[4] J.C. Newman, A.C. Taylor, R.W. Barnwell, P.A. Newman, G.J.W. Hou, Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations, J. Aircraft 36 (1) (1999) 87–96.

[5] J.P.C. Kleijnen, Sensitivity analysis and related analyses: a review of some statistical techniques, J. Statist. Comput. Simul. 57 (1997) 111–142.

[6] K.K. Choi, N.H. Kim, Structural Sensitivity Analysis and OptimizationLinear Systems, vol. 1, Springer-Verlag, New York, 2004.

[7] K.K. Choi, N.H. Kim, Structural Sensitivity Analysis and OptimizationNonlinear Systems and Applications, vol. 2, Springer-Verlag, New York, 2004.

[8] J. Sobieszczanski-Sobieski, Sensitivity of complex, internally coupled systems, AIAA J. 28 (1) (1990) 153–160.

[9] A.A. Guinta, Sensitivity analysis for coupled aero-structural systems, Technical Report TM-1999-209367, NASA, 1999.

[10] G.J.-W. Hou, A. Satyanarayana, Analytical sensitivity analysis of a static aeroelastic wing, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4824, 6–8 September 2000, Long Beach, CA.

[11] K. Maute, M. Nikbay, C. Farhat, Analytically based sensitivity analysis and optimization of nonlinear aeroelastic systems, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000–4825, 6–8 September 2000, Long Beach, CA.

[12] V.V. Saurin, Shape design sensitivity analysis for fracture conditions, Comput. Struct. 76 (2000) 399–405.

[13] E. Taroco, Shape sensitivity analysis in linear elastic fracture mechanics, Comput. Methods Appl. Mech. Engrg. 188 (2000) 697–712.

[14] R.A. Feijóo, C. Padra, R. Saliba, E. Taroco, M.J. Vénere, Shape sensitivity analysis for energy release rate evaluation and its application to the study of three-dimensional cracked bodies, Comput. Methods Appl. Mech. Engrg. 188 (2000) 649–664.

[15] N.H. Kim, J. Dong, K.K. Choi, N. Vlahopoulos, Z.-D. Ma, M.P. Castanier, C. Pierre, Design sensitivity analysis for sequential structural-acoustic problems, J. Sound Vib. 263 (3) (2003) 569–591.

[16] N.H. Kim, J. Dong, K.K. Choi, Energy flow analysis and design sensitivity of structural problems at high frequencies, J. Sound Vib. 269 (1–2) (2004) 213–250.

[17] K.K. Choi, I. Shim, S. Wang, Design sensitivity analysis of structure-induced noise and vibration, J. Vib. Acoust. 119 (1997) 173–179.

[18] M. Papadrakakis, Y. Tsompanakis, E. Hinton, J. Sienz, Advanced solution methods in topology optimization and shape sensitivity analysis, Engrg. Comput. 13 (5) (1996) 57–90.

[19] K. Maute, M. Nikbay, C. Farhat, Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method, Int. J. Numer. Methods Engrg. 56 (6) (2003) 911–933.

[20] T.S. Kim, J.E. Kim, Y.Y. Kim, Parallelized structural topology optimization for eigenvalue problems, Int. J. Solid Struct. 41 (9–10) (2004) 2623–2641.

[21] S.J. Leary, A. Bhaskar, A.J. Keane, Global approximation and optimization using adjoint computational fluid dynamics codes, AIAA J. 42 (3) (2004) 631–641.

[22] O.O. Storaasli, D.T. Nguyen, M. Baddourah, J. Qin, Computational mechanics analysis tools for parallel-vector supercomputers, in: 34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, La Jolla, CA, USA, 19–22 April 1993, AIAA, pp. 772–778.

[23] B.C. Watson, A.K. Noor, Sensitivity analysis of frictional contact/impact response on distributed-memory computers, Comput. Struct. 65 (6) (1997) 881–891.

[24] I. Takewaki, K. Uetani, Efficient redesign of damped large structural systems via domain decomposition with exact dynamic condensation, Comput. Methods Appl. Mech. Engrg. 178 (3) (1999) 367–382.

[25] M. Aleksa, S. Russenschuck, C. Vollinger, Parametric quadrilateral meshes for the design and optimization of superconducting magnets, IEEE Trans. Appl. Supercond. 12 (1) (2002) 1404–1407.

[26] L.A. Schemit, Y.C. Lai, Structural optimization based on preconditioned conjugate gradient analysis methods, Int. J. Numer. Methods Engrg. 37 (6) (1994) 943–964.

[27] C. Vuik, Fast iterative solvers for the discretized incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 22 (1996) 195–210.

[28] C. Vuik, A. Segal, J.A. Meijerink, An efficient preconditioned cg method for the solution of a class of layered problems with extreme contrasts in the coefficients, J. Comput. Phys. 152 (1999) 385–403.

[29] J. Frank, C. Vuik, On the construction of deflation-based preconditioners, SIAM J. Sci. Comput. 23 (2) (2001) 442–462.

[30] M.A. Akgün, J.H. Garcelon, R.T. Haftka, Fast exact linear and nonlinear structural reanalysis and the Sherman–Morrison–Woodbury formulas, Int. J. Numer. Methods Engrg. 50 (7) (2001) 1587–1606.

[31] B.G. Yoon, A.D. Belegundu, Iterative methods for design sensitivity analysis, AIAA J. 26 (11) (1988) 1413–1415.

[32] U. Kirsch, S. Liu, Structural reanalysis for general layout modifications, AIAA J. 35 (1997) 382–388.

[33] S. Oral, An improved semianalytical method for sensitivity analysis, Struct. Optim. 11 (1996) 67–69.

[34] H.R.E.M. Hörnlein, Effiziente semi-analytische gradientenberechnung in der strukturoptimierung, Z. Angew. Math. Mech. 81 (2000) s669–s670.

[35] U. Kirsch, Design-oriented Analysis: a Unified Approach, Kluwer, 2002.

[36] R.T. Haftka, Semi-analytical static nonlinear structural sensitivity analysis, AIAA J. 31 (1993) 1307–1312.

[37] B.Y. Lee, Consideration of body forces in axisymmetric design sensitivity analysis using the BEM, Comput. Struct. 61 (4) (1996) 587–596.

[38] B.Y. Lee, Direct differentiation formulation for boundary element shape sensitivity analysis of axisymmetric elastic solids, Int. J. Solid Struct. 34 (1) (1997) 99–112.

[39] N.H. Kim, K.K. Choi, Design sensitivity analysis and optimization of nonlinear transient dynamics, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4905, 6–8 September 2000, Long Beach, CA.

[40] F. Navarrina, S. López-Fontán, I. Colominas, E. Bendito, M. Casteleiro, High order shape design sensitivities: a unified approach, Comput. Methods Appl. Mech. Engrg. 188 (2000) 681–696.

[41] K.K. Choi, W. Duan, Design sensitivity analysis and shape optimization of structural components with hyperelastic material, Comput. Methods Appl. Mech. Engrg. 187 (2000) 219–243.

[42] S. Mukherjee, X. Shi, Y.X. Mukherjee, Internal variables and their sensitivities in three-dimensional linear elasticity by the boundary contour method, Comput. Methods Appl. Mech. Engrg. 187 (2000) 289–306.

[43] D.A. Tortorelli, R.B. Haber, S.C.-Y. Lu, Adjoint sensitivity analysis for nonlinear dynamic thermoelastic systems, AIAA J. 29 (2) (1991) 253–263.

[44] J.S. Arora, T.-H. Lee, J.B. Cardoso, Structural shape sensitivity analysis: relationship between material derivative and control volume approaches, AIAA J. 30 (6) (1992) 1638–1648.

[45] J.S. Arora, An exposition of the material derivative approach for structural shape sensitivity analysis, Comput. Methods Appl. Mech. Engrg. 105 (1993) 41–62.

[46] D.G. Phelan, R.B. Haber, Sensitivity analysis of linear elastic systems using domain parametrization and a mixed mutual energy principle, Comput. Methods Appl. Mech. Engrg. 77 (1989) 31–59.

[47] N.H. Kim, Y. Chang, Eulerian shape design optimization with fixed grid, in: 44th AIAA/ASME/ASCE/ASC Structure, Structural Dynamics, and Materials Conference, AIAA 2003–1634, 7–10 April 2003, Norfolk, VA.

[48] C. Bischof, A. Carle, Automatic differentiation—principles, tools, and applications in sensitivity analysis, in: K. Chan, S. Tarantola, F. Campolongo (Eds.), Second International Symposium on Sensitivity Analysis of Model Output, 19–22 April 1998, Venice, Ca'Dolfin, pp. 33–36.

[49] P.C. Pandey, P. Bakshi, Analytical response sensitivity computation using hybrid finite elements, Comput. Struct. 71 (1999) 525–534.

[50] D.W. Stillman, Design sensitivity analysis for structures using explicit time integration, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4906, 6–8 September 2000, Long Beach, CA.

[51] J.S. Moita, J. Infanta Barbosa, C.M. Mota Soares, C.A. Mota Soares, Sensitivity analysis and optimal design of geometrically non-linear laminated plates and shells, Comput. Struct. 76 (2000) 407–420.

[52] R. Fortunier, F. Thevon, J.D. Beley, Parametric finite element analysis: a new design method to shorten time to market, in: Proceedings of the EUROMAT Conference, 1997, Maastricht, p. 1524.

[53] M.J. Poldneff, I.S. Rai, J.S. Arora, Implementation of design sensitivity analysis for nonlinear structures, AIAA J. 31 (11) (1993) 2137–2142.

[54] S. Kibsgaard, Sensitivity analysis—the basis for optimization, Int. J. Numer. Methods Engrg. 34 (1992) 901–932.

[55] C.H. Bischof, H.M. Bücker, B. Lang, A. Rasch, J.W. Risch, Extending the functionality of the general-purpose finite element package sepran by automatic differentiation, Int. J. Numer. Methods Engrg. 58 (2003) 2225–2238.

[56] L.L. Sherman, A.C. Taylor, L.L. Green, P.A. Newman, G.J.-W. Hou, V.M. Korivi, First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods, in: 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 94-4262, 7–9 September 1994, Panama City Beach, FL.

[57] I. Ozaki, F. Kimura, M. Berz, Higher-order sensitivity analysis of finite element method by automatic differentiation, Comput. Mech. 16 (1995) 223–234.

[58] B.A. Wujek, J.E. Renaud, Automatic differentiation for more efficient system analysis and optimization, Engrg. Optim. 31 (1998) 101–139.

[59] C. Bischof, A. Carle, P. Khademi, A. Mauer, The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs, IEEE Comput. Sci. Engrg. 3 (1996) 18–32.

[60] A. Griewank, D. Juedes, J. Utke, ADOL-C, a package for the automatic differentiation of algorithms written in C/C++, TOMS 22 (2) (1996) 131–167.

[61] D. Shiriaev, A. Griewank, ADOL-F: Automatic differentiation of Fortran codes, Comput. Differentiat. Tech. Appl. Tools 1 (1996) 375–384.

[62] J.R.R.A. Martins, P. Sturdza, J.J. Alonso, The connection between the complex-step derivative approximation and algorithmic differentiation, AIAA 2001–0921, in: Proceedings of the 39th Aerospace Sciences Meeting, Reno, NV, January 2001.

[63] J.N. Lyness, C.B. Moler, Numerical differentiation of analytic functions, SIAM J. Numer. Anal. 4 (1967) 202–210.

[64] J.N. Lyness, Numerical algorithms based on the theory of complex variables, in: Proceedings of the ACM 22nd Notational Conference, Thomas Book, Co., Washington, DC, 1967, pp. 124–134.

[65] W. Squire, G. Trapp, Using complex variable to estimate derivatives of real functions, SIAM Rev. 40 (1) (1998) 110–112.

[66] J.C. Newman, D.L. Whitfield, W.K. Anderson, A step-size independent approach for multidisciplinary sensitivity analysis and design optimization, in: 17th AIAA Applied Aerodynamics Conference, 28 June–1 July 1999, pp. 12–22.

[67] J.R.R.A. Martins, I.M. Kroo, J.J. Alonso, An automated method for sensitivity analysis using complex variables, in: AIAA 38th Aerospace Sciences Meeting and Exhibit, Reno NV, AIAA Paper 2000-0689, 2000.

[68] G. Bugeda, E. Onate, Optimum aerodynamic shape design including mesh adaptivity, Int. J. Numer. Methods Fluids 20 (1995) 915–934.

[69] K.K. Choi, S.-L. Twu, Equivalence of continuum and discrete methods of shape design sensitivity analysis, AIAA J. 27 (10) (1989) 1418–1424.

[70] S.K. Nadarajah, A. Jameson, A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization, in: AIAA 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper No. 2000-0667, January 2000.

[71] M.D. Gunzburger, Sensitivities, adjoints and flow optimization, Int. J. Numer. Methods Fluids 31 (1999) 53–58.

[72] B. Barthelemy, R.T. Haftka, Accuracy analysis of the semi-analytical method for shape sensitivity, in: Proceedings of AIAA/ASME/ASCE/ASC 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, VA, 18–20 April 1988, pp. 88–2284.

[73] B. Barthelemy, C.T. Chon, R.T. Haftka, Accuracy problems associated with semi-analytical derivatives of static response, in: Presented at the First World Congress on Computational Mechanics, Austin, TX, September 1986, Finite Element Anal. Des. 4 (1988) 249–265.

[74] H.P. Mlejnek, Accuracy of semi-analytical sensitivities and its improvement by the natural method, Struct. Optim. 4 (1992) 128–131.

[75] N. Olhoff, J. Rasmussen, E. Lund, A method of "exact" numerical differentiation for error elimination in finite-element-based semi-analytical shape sensitivity analyses, Mech. Struct. Mach. 21 (1993) 1–66.

[76] F. van Keulen, H. de Boer, Rigorous improvement of semi-analytical design sensitivities by exact differentiation of rigid body motions, Int. J. Numer. Methods Engrg. 42 (1998) 71–91.

[77] H. de Boer, F. vanKeulen, Refined semi-analytical design sensitivities, Int. J. Solid Struct. 37 (46–47) (2000) 6961–6980.

[78] K. Vervenne, Design sensitivity analysis, Technical report, Delft University of Technology, LTM 1234, 2000.

[79] J. Borggaard, A. Verma, On efficient solutions to the continuous sensitivity equation using structured automatic differentiation, SIAM J. Sci. Comput. 22 (1) (2000) 39–62.

[80] A. Sandu, G.R. Charmichael, F.A. Potra, Sensitivity analysis for atmospheric chemistry via automatic differentiation, Paper presented at SAMO 95, Belgirate, Italy, 25–27 September 1995.

[81] P. Krishnaswami, Symbolic computing and automatic differentiation for sensitivity analysis, in: Optimization in Industry—1997, ASME, 1997, pp. 103–108.

[82] T. Burczynski, J.H. Kane, C. Balakrishna, Comparison of shape design sensitivity analysis formulations via material derivative-adjoint variable and implicit differentiation techniques for 3-d and 2-d curved boundary element, Comput. Methods Appl. Mech. Engrg. 142 (1997) 89–109.

[83] J. Borggaard, J. Burns, A pde sensitivity equation method for optimal aerodynamical design, J. Comput. Phys. 136 (1997) 366–384.

[84] K. Dems, R.T. Haftka, Two approaches to sensitivity analysis for shape variation of structures, Mech. Struct. Mach. 16 (1989) 501–522.

[85] T. Lindby, J.L.T. Santos, 2-d and 3-d shape optimization using mesh velocities to integrate analytical sensitivities with associative cad, Struct. Optim. 13 (1997) 213–222.

[86] Z. Erman, R.T. Fenner, Three-dimensional design sensitivity analysis using a boundary integral approach, Int. J. Numer. Methods Engrg. 40 (1997) 637–654.

[87] A. Carle, M. Fagan, L.L. Green, Preliminary results from the application of automated adjoint code generation to CFL3D, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, volume Part 2, AIAA 98-4807, 2–4 September 1998, pp. 807–817.

[88] J.L. Walsh, K.C. Young, F.J. Tarzanin, J.E. Hirsh, D.E. Young, Optimization issues with complex rotorcraft comprehensive analysis, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, volume Part 2, AIAA 98-4889, 2–4 September 1998.

[89] H. Hu, sensitivity derivatives of a rotor CFD code via automatic differentiation, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, volume Part 1, AIAA 98-4724, pages 159–164, September 2–4 1998.

[90] M.A. Akgün, R.T. Haftka, K.C. Wu, J.L. Walsh, Sensitivity of lumped constraints using the adjoint method, AIAA J. 39 (3) (2001) 511–516.

[91] B.M. Barthelemy, R.T. Haftka, G.A. Cohen, Physically based sensitivity derivatives for structural analysis programs, Comput. Mech. 4 (6) (1989) 465–476.

[92] K.H. Chang, K.K. Choi, C.S. Tsai, B.S. Choi, X.M. Yu, Design sensitivity analysis and optimization tool (DSO) for shape design applications, Comput. System Engrg. 6 (2) (1995) 151–175.

[93] T.J.R. Hughes, The Finite Element Method, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[94] E.J. Haug, K.K. Choi, V. Komkov, Design Sensitivity Analysis of Structural Systems, Academic Press, New York, 1986.

[95] W.H. Greene, R.T. Haftka, Computational aspects of sensitivity calculations in transient structural analysis, Comput. Struct. 32 (1989) 433–443.

[96] M.A. Kramer, J.M. Calo, H. Rabitz, An improved computational method for sensitivity analysis: Green's function method with aim, Appl. Math. Model. 5 (1981) 432–441.

[97] R.T. Haftka, Z. Gürdal, Elements of Structural Optimization, second revised edition., Kluwer Academic Publishers, 1990.

[98] M.J. Poldneff, J.S. Arora, Design sensitivity analysis in dynamic thermoviscoelasticity with implicit integration, Int. J. Solid Struct. 33 (4) (1996) 577–594.

[99] S. Cho, K.K. Choi, Design sensitivity analysis and optimization of non-linear transient dynamics. Part I—Sizing design, Int. J. Numer. Methods Engrg. 48 (2000) 351–373.

[100] N.H. Kim, K.K. Choi, Design sensitivity analysis and optimization of nonlinear transient dynamics, Mech. Struct. Mach. 29 (3) (2001) 351–371.

[101] R.T. Haftka, D.S. Malkus, Calculation of sensitivity derivatives in thermal problems by finite differences, Int. J. Numer. Methods Engrg. 17 (1981) 1811–1821.

[102] G.D. Pollock, A.K. Noor, Sensitivity analysis of the contact/impact response of composite structures, Comput. Struct. 61 (2) (1996) 251–269.

[103] E.J. Haug, N.K. Mani, P. Krishnaswami, in: E.J. Haug (Ed.), Design Sensitivity Analysis and Optimization of Dynamically Driven Systems, Springer, Berlin, 1984, pp. 555–636.

[104] L. Karaoğlan, A.K. Noor, Dynamic sensitivity analysis of frictional contact/impact response of axisymmetric composite structures, Comput. Methods Appl. Mech. Engrg. 128 (1995) 169–190.

[105] R.T. Haftka, Techniques for thermal sensitivity analysis, Int. J. Numer. Methods Engrg. 17 (1981) 71–80.

[106] P. Michaleris, D.A. Tortorelli, C.A. Vidal, Tangent operators and design sensitivity formulations for transient non-linear coupled problems with applications to elastoplasticity, Int. J. Numer. Methods Engrg. 37 (1994) 2471–2499.

[107] S. Chen, J.M. Hansen, D.A. Tortorelli, Unconditionally energy stable implicit time integration: application to multibody system analysis and design, Int. J. Numer. Methods Engrg. 48 (2000) 791–822.

[108] T.V. Nguyen, A. Devgan, O.J. Nastov, D.W. Winston, Transient sensitivity computation in controlled explicit piecewise linear simulation, IEEE Trans. Comp.-Aided Des. Integr. Circ. Syst. 19 (1) (2000) 98–110.

[109] C.C. Wu, J.S. Arora, Design sensitivity analysis and optimization of nonlinear structural response using incremental procedure, AIAA J. 25 (1987) 1118–1125.

[110] C.C. Wu, J.S. Arora, Design sensitivity analysis of non-linear buckling load, Comput. Mech. 3 (1988) 129–140.

[111] E. Lund, N. Olhoff, Shape design sensitivity analysis of eigenvalues using "exact" numerical differentiation of finite element matrices, Struct. Optim. 8 (1994) 52–59.

[112] A.M. Spence, R. Celi, Efficient sensitivity analysis for rotary-wing aeromechanical problems, AIAA J. 32 (12) (1994) 2337–2344.

[113] W.H. Zhang, M. Domaszewski, Efficient sensitivity analysis and optimization of shell structures by the ABAQUS code, Struct. Optim. 18 (1999) 173–182.

[114] J.W. Langelaan, E. Livne, Analytic sensitivities and design oriented structural analysis for airplane fuselage shape synthesis, Comput. Struct. 62 (3) (1997) 505–519.

[115] G. Cheng, Y. Gu, Y. Zhou, Accuracy of semi-analytic sensitivity analysis, Finite Element Anal. Des. 6 (1989) 113–128.

[116] P. Pedersen, G. Cheng, J. Rasmussen, On accuracy problems for semi-analytical sensitivity analyses, Mech. Struct. Mach. 17 (1989) 373–384.

[117] P. Fenyes, R.V. Lust, Error analysis of semianalytic displacement derivatives for shape and sizing variables, AIAA J. 29 (1991) 271–279.

[118] N. Olhoff, J. Rasmussen, Study of inaccuracy in semi-analytical sensitivity analysis—a model problem, Struct. Optim. 3 (1991) 203–213.

[119] G. Cheng, N. Olhoff, Rigid body motion test against error in semi-analytical sensitivity analysis, Comput. Struct. 46 (1993) 515–527.

[120] F. van Keulen, K. Vervenne, H. de Boer, Accuracy improvement of semi-analytical design sensitivities—an overview of recent developments, in: O.M. Querin (Ed.), Proceedings of 3rd ASMO-UK/ISSMO Conference Harrogate, North Yorkshire, 9–10 July 2001, Engineering Design Optimization, Product and Process Improvement, University Press Leeds, 2001, pp. 227–236.

[121] G. Cheng, Y. Gu, X. Wang, Improvement of semi-analytical analysis and mcads, in: H.A. Eschenhauer, C. Mattheck, N. Olhoff (Eds.), Engineering Optimization in Design Processes, Springer, Berlin, Heidelberg, New York, 1991, pp. 211–223.

[122] H. de Boer, F. van Keulen, Error analysis of refined semi-analytical design sensitivities, Struct. Optim. 14 (1997) 242–247.

[123] J. Argyris, H.P. Mlejnek, Die Methode der Finiten Elemente, vol. I: Verschiebungsmethode in der Statik, Braunschweig, Vieweg, 1986 (in German).

[124] E. Hinton, J. Sienz, S.M.B. Afonso, Experiences with Olhoff's "exact" semi-analytical sensitivity algorithm, in: N. Olhoff, G.I.N. Rozvany (Eds.), WCSMO-1, First World Congress of Structural and Multidisciplinary Optimization, Pergamon, 1995, pp. 41–46.

[125] F. van Keulen, H. de Boer, Refined semi-analytical design sensitivities for buckling, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, volume Part 1, AIAA 98-4761, 2–4 September 1998, pp 420–429.

[126] H. de Boer, F. van Keulen, Refined second order semi-analytical design sensitivities, in: Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization, 17–21 May 1999, Buffalo, New York, CD-ROM (Paper 12-SAM-4).

[127] E. Parente, L.E. Vaz, Improvement of semi-analytical design sensitivities of nonlinear structures using equilibrium relations, Int. J. Numer. Methods Engrg. 50 (9) (2001) 2127–2142.

[128] J.E. Bernard, S.K. Kwon, J.A. Wilson, Differentiation of mass and stiffness matrices for high order sensitivity calculations in finite element-based equilibrium problems, Trans. ASME 115 (1993) 829–832.

[129] P. Bakshi, P.C. Pandey, Semi-analytic sensitivities using hybrid finite elements, Comput. Struct. 77 (2000) 201–213.

[130] K. Vervenne, F. van Keulen, Accuracy improvement of semi-analytical design sensitivities by Laplacian smoothing, in: 42nd Structures, Structural Dynamics and Materials Conference and Exhibit, AIAA 2001-1497, A01-25415, Seattle, WA, 16–19 April 2001, AIAA.

[131] W.-H. Zhang, P. Beckers, C. Fleury, A unified parametric design approach to structural shape optimization, Int. J. Numer. Methods Engrg. 38 (1995) 2283–2292.

[132] C.H. Tseng, J.S. Arora, Numerical verification of design sensitivity analysis, AIAA J. 27 (1) (1989) 117–119.

[133] K. Tai, R.T. Fenner, Numerical study of some approaches to shape design sensitivity analysis using boundary elements, J. Strain Anal. 31 (5) (1996) 361–369.

[134] J. Borggaard, J. Burns, Asymptotically consistent gradients in optimal design, in: Multidisciplinary Design Optimization, State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, Hampton, Virginia, 13–16 March 1995, pp. 303–314.

[135] J.H. Kane, K.G. Prasad, Boundary formulations for sensitivity analysis without matrix derivatives, AIAA J. 31 (9) (1993) 1731–1734.

[136] L.A. Godoy, E.O. Taroco, Design sensitivity of post-buckling states including material constraints, Comput. Methods Appl. Mech. Engrg. 188 (2000) 665–679.

[137] H. de Boer, F. van Keulen, Improved semi-analytic design sensitivities for a linear and finite rotation shell element, in: W. Gutkowski, Z. Mróz (Eds.), WCSMO-2, Second World Congress of Structural and Multidisciplinary Optimization, 1997, pp. 199–204.

[138] D.E. Smith, D.A. Tortorelli, C.L. Tucker, Optimal design for polymer extrusion. Part II: Sensitivity analysis for weakly-coupled nonlinear steady-state systems, Comput. Methods Appl. Mech. Engrg. 167 (1998) 303–323.

[139] M.S. Joun, S.M. Hwang, Die shape optimal design in three-dimensional shape metal extrusion by the finite element method, Int. J. Numer. Methods Engrg. 41 (1998) 311–335.

[140] H. Møller, E. Lund, Shape sensitivity analysis of strongly coupled fluid-structure interaction problems, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4823, 6–8 September 2000, Long Beach, CA.

[141] L.A. Jakobsen, H. Møller, E. Lund, Sensitivity analysis of convection dominated steady 2d fluid flow using supg fem, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000–4822, 6–8 September 2000, Long Beach, CA.

[142] G. Cheng, N. Olhoff, New method of error analysis and detection in semi-analytic sensitivity analysis, in: G.I.N. Rozvany (Ed.), Lecture Notes, NATO/DFG/ASI Optimization of Large Structural Systems, vol. 1, 23 September–4 October 1991, pp. 234–267.

[143] I. Ozaki, T. Terano, Applying an automatic differentiation technique to sensitivity analysis in design optimization problems, Finite Element Anal. Des. 14 (1993) 143–151.

[144] C. Faure, C. Ducal, Automatic differentiation for sensitivity analysis, a test case, in: K. Chan, S. Tarantola, F. Campolongo (Eds.), Second International Symposium on Sensitivity Analysis of Model Output, Venice, Ca'Dolfin, 19–22 April 1998, pp. 107–110.

[145] H. Hu, Application of an automatic differentiation method to a 2d Navier–Stokes CFD code, Comput. Methods Appl. Mech. Engrg. 156 (1998) 179–183.

[146] A.C. Limache, E.M. Cliff, Aerodynamic sensitivity theory for rotary stability derivatives, in: AIAA Atmospheric Flight Mechanics Conference, Portland, OR, 9–11 August 1999, AIAA 99-4313.

[147] L.B. Rall, G.F. Corliss, An introduction to automatic differentiation, Comput. Differentiat.: Tech. Appl. Tools (1996) 1–18.

[148] C.-J. Chen, K.K. Choi, Continuum approach for second-order shape design sensitivity of three-dimensional elastic solids, AIAA J. 32 (10) (1994) 2099–2107.

[149] F.-J. Barthold, E. Stein, A continuum mechanical-based formulation of the variational sensitivity analysis in structural optimization. Part I: Analysis, Struct. Optim. 11 (1996) 29–42.

[150] H.-Y. Hwang, K.K. Choi, K.-H. Chang, Second-order shape design sensitivity using p-version finite element analysis, Struct. Optim. 14 (1997) 91–99.

[151] P. Kolakowski, J. Holnicki-Szulc, Sensitivity analysis of truss structures (virtual distortion method approach), Int. J. Numer. Methods Engrg. 43 (1998) 1085–1108.

[152] H.W. Kim, D.S. Bae, K.K. Choi, Configuration design sensitivity analysis of kinematic responses of mechanical systems, Mech. Struct. Mach. 27 (2) (1999) 203–215.

[153] É. Turgeon, D. Pelletier, J. Borggaard, A general continuous sensitivity equation formulation for complex flows, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4732, 6–8 September 2000, Long Beach, CA.

[154] É. Turgeon, D. Pelletier, J. Borggaard, B. Ozell, A continuous sensitivity equation method for flows with temperature dependent properties, in: 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA 2000-4821, 6–8 September 2000, Long Beach, CA.

[155] E. Rohan, J.R. Whiteman, Shape optimization of elasto-plastic structures and continua, Comput. Methods Appl. Mech. Engrg. 187 (2000) 261–288.

[156] M. Souli, J.P. Zolesio, A. Ouahsine, Shape optimization for non-smooth geometry in two dimensions, Comput. Methods Appl. Mech. Engrg. 188 (2000) 109–119.

[157] J.A. Brandon, Second-order design sensitivities to assess the applicability of sensitivity analysis, AIAA J. 29 (1) (1991) 135–139.

[158] M.I. Friswell, Calculation of second and higher order eigenvector derivatives, J. Guidance 18 (4) (1995) 919–921.

[159] L.A. Godoy, E.O. Taroco, R.A. Feijoo, Second-order sensitivity analysis in vibration and buckling problems, Int. J. Numer. Methods Engrg. 37 (1994) 3999–4014.

[160] J. Borggaard, D. Pelletier, On adaptive shape sensitivity calculations for optimal design, Comput. Methods Appl. Mech. Engrg., submitted for publication.

[161] J. Borggaard, D. Pelletier, Observations in adaptive refinement strategies for optimal design, in: J. Borggaard, J. Burns, E. Cliff, S. Schreck (Eds.), Computational Methods for Optimal Design and Control, Birkhauser, 1998, pp. 59–76.

[162] J. Borggaard, D. Pelletier, Optimal shape design in forced convection using adaptive finite elements, AIAA 98-0908, in: AIAA 36th Aerospace Sciences Meeting and Exhibit, Reno NV, 1998.

[163] N.H. Kim, K.K. Choi, J.S. Chen, Y.H. Park, Meshless shape design sensitivity analysis and optimization for contact problem with friction, Comput. Mech. 25 (2-3) (2000) 157–168.

[164] N.H. Kim, K.K. Choi, J.S. Chen, Shape design sensitivity analysis and optimization of elasto-plasticity with frictional contact, AIAA J. 38 (9) (2000) 1742–1753.

[165] N.H. Kim, K. Yi, K.K. Choi, A material derivative approach in design sensitivity analysis of 3-d contact problems, Int. J. Solid Struct. 39 (8) (2002) 2087–2108.

[166] N.H. Kim, Y.H. Park, K.K. Choi, Optimization of a hyperelastic structure with multibody contact using continuum-based shape design sensitivity analysis, Struct. Optim. 21 (3) (2001) 196–208.

[167] N.H. Kim, K.K. Choi, J.S. Chen, Die shape design optimization of sheet metal stamping process using meshfree method, Int. J. Numer. Methods Engrg. 51 (12) (2001) 1385–1405.

[168] T. Ting, Design sensitivity analysis of structural frequency response, AIAA J. 31 (10) (1992) 1965–1967.

[169] Y. Zhang, D.T. Nguyen, J.W. Hou, An alternative formulation for design sensitivity analysis of linear structural dynamic systems, Comput. Struct. 44 (3) (1992) 689–692.

[170] A.G. Nalecz, J. Wicher, Design sensitivity analysis of mechanical systems in frequency domain, J. Sound Vib. 120 (3) (1988) 517–526.

[171] R.M. Lin, M.K. Lim, Structural sensitivity analysis via reduced-order analytical model, Comput. Methods Appl. Mech. Engrg. 121 (1995) 345–359.

[172] R.M. Lin, H. Du, J.H. Ong, Sensitivity based method for structural dynamic model improvement, Comput. Struct. 47 (3) (1993) 349–369.

[173] A.L. Andrew, R.C.E. Tan, Computation of derivatives of repeated eigenvalues and corresponding eigenvectors by simultaneous iteration, AIAA J. 34 (10) (1996) 2214–2216.

[174] A.L. Andrew, R.C.E. Tan, Computation of mixed partial derivatives of eigenvalues and eigenvectors by simultaneous iteration, Commun. Numer. Methods Engrg. 15 (1999) 641–649.

[175] C. Cardani, P. Mantegazza, Calculation of eigenvalue and eigenvector derivatives for algebraic flutter and divergence eigenproblems, AIAA J. 17 (4) (1979) 408–412.

[176] T. Chen, Design sensitivity analysis for repeated eigenvalues in structural design, AIAA J. 31 (12) (1993) 2347–2350.

[177] K.K. Choi, E.J. Haug, H.G. Seong, An iterative method for finite dimensional structural optimization problems with repeated eigenvalues, Int. J. Numer. Methods Engrg. 19 (1983) 93–112.

[178] R.L. Dailey, Eigenvector derivatives with repeated eigenvalues, AIAA J. 27 (4) (1989) 486–491.

[179] E.J. Haug, K.K. Choi, Systematic occurrence of repeated eigenvalues in structural optimization, J. Optim. Theory Appl. 38 (2) (1982) 251–274.

[180] R.M. Lin, Z. Wang, M.K. Lim, A practical algorithm for the efficient computation of eigenvector sensitivities, Comput. Methods Appl. Mech. Engrg. 130 (1996) 355–367.

[181] R.L. Fox, M.P. Kapoor, Rates of change of eigenvalues and eigenvectors, AIAA J. 6 (12) (1968) 2426–2429.

[182] M.S. Jankovic, Exact *n*th derivatives of eigenvalues and eigenvectors, J. Guid. Control Dynam. 17 (1) (1994) 136–144.

[183] K. Kim, D.V. Wallerstein, Modal design sensitivities for multiple eigenvalues, Comput. Struct. 29 (5) (1988) 755–762.

[184] H.C. Mateus, H.C. Rodrigues, C.M. Mota Soares, C.A. Mota Soares, Sensitivity analysis and optimization of thin laminated structures with a nonsmooth eigenvalue based criterion, Struct. Optim. 14 (1997) 219–224.

[185] W.C. Mills-Curran, Calculation of eigenvector derivatives for structures with repeated eigenvalues, AIAA J. 26 (7) (1988) 867–871.

[186] W.C. Mills-Curran, Comment on "eigenvector derivatives with repeated eigenvalues", AIAA J. 28 (10) (1990) 1846.

[187] N. Olhoff, E. Lund, A.P. Seyranian, Sensitivity analysis and optimization of multiple eigenvalues in structural design problems, in: Proceedings of the AIAA/NASA/USAF/ISMO Symposium on Multidisciplinary Analysis and Optimization, Part 1, 1994, AIAA, pp. 625–640.

[188] V.F. Poterasu, Approximate method to compute the eigenvalues and eigensensitivities of mechanical systems, Comput. Methods Appl. Mech. Engrg. 109 (1993) 183–192.

[189] A.R. Robinson, J.F. Harris, Improving approximate eigenvalues and eigenvectors, in: Proceedings of the American Society of Civil Engineers, EM, J. Engrg. Mech. Div. EM 2 (1971) 457–475.

[190] L.C. Rogers, Derivatives of eigenvalues and eigenvectors, AIAA J. 8 (5) (1970) 943–944.

[191] C.S. Rudisill, Y.Y. Chu, Numerical methods for evaluating the derivatives of eigenvalues and eigenvectors, AIAA J. 13 (6) (1975) 834–836.

[192] A.P. Seyranian, E. Lund, N. Olhoff, Multiple eigenvalues in structural optimization problems, Struct. Optim. 8 (1994) 207–227.

[193] R.C.E. Tan, A.L. Andrew, F.M.L. Hong, Iterative computation of second-order derivatives of eigenvalues and eigenvectors, Commun. Numer. Methods Engrg. 10 (1994) 1–9.

[194] W.H. Wittrick, Rates of change of eigenvalues with reference to buckling and vibration problems, J. R Aeronautic. Soc. 66 (1966) 590–591.

[195] R.B. Nelson, Simplified calculation of eigenvector derivatives, AIAA J. 14 (9) (1976) 1201–1205.

[196] I.U. Ojalvo, Efficient computation of modal sensitivities for systems with repeated frequencies, AIAA J. 26 (3) (1988) 361–366.

[197] J.-G. Beliveau, S. Cogan, G. Lallement, F. Ayer, Iterative least-squares calculation for modal eigenvector sensitivity, AIAA J. 34 (2) (1996) 385–391.

[198] B. Lallemand, P. Level, H. Duveau, B. Mahieux, Eigensolutions sensitivity analysis using a sub-structuring method, Comput. Struct. 71 (1999) 257–265.

[199] T. Ting, Accelerated subspace iteration for eigenvector derivatives, AIAA J. 30 (8) (1992) 2114–2118.

[200] O. Zhang, A. Zerva, Accelerated iterative procedure for calculation eigenvector derivatives, AIAA J. 35 (2) (1997) 340–348.

[201] M.S. Zarghamee, Optimum frequency of structures, AIAA J. 6 (4) (1968) 749–750.

[202] I. Grindeanu, N.H. Kim, K.K. Choi, J.S. Chen, Cad-based shape optimization using a meshfree method, Concurrent Engrg.: Res. Appl. 10 (1) (2002) 55–66.

[203] Y.H. Park, K.K. Choi, Configuration design sensitivity analysis of nonlinear structural systems with elastic material, Mech. Struct. Mach. 24 (2) (1996) 217–255.

[204] J.S. Park, K.K. Choi, Design sensitivity analysis of critical load factor for nonlinear structural systems, Comput. Struct. 36 (5) (1990) 823–838.

[205] K.K. Choi, J.S. Park, Optimal design of nonlinear structural systems with critical loads, SAE Trans. 99 (Sect6) (1990) 1141–1153.

[206] A. Eriksson, Fold lines for sensitivity analyses in structural instability, Comput. Methods Appl. Mech. Engrg. 114 (1994) 77–101.

[207] T.S. Kwon, B.C. Lee, W.J. Lee, An approximation technique for design sensitivity analysis of the critical load in non-linear structures, Int. J. Numer. Methods Engrg. 45 (1999) 1727–1736.

[208] N. Olhoff, S.H. Rasmussen, On single and bimodal optimum buckling loads of clamped columns, Int. J. Solid Struct. 13 (1977) 605–614.

[209] N.H. Kim, K.K. Choi, J.S. Chen, Shape optimization of finite deformation elastoplasticity using continuum-based design sensitivity formulation, Comput. Struct. 79 (20–21) (2001) 1959–1976.

[210] F. Bobaru, S. Mukherjee, Shape sensitivity analysis and shape optimization in planar elasticity using the element-free galerkin method, Comput. Methods Appl. Mech. Engrg. 190 (32–33) (2001) 4319–4337.

[211] K.K. Choi, N.H. Kim, Design optimization of springback in a deepdrawing process, AIAA J. 40 (1) (2002) 147–153.

[212] N.H. Kim, K.K. Choi, J.S. Chen, M.E. Botkin, Meshfree analysis and design sensitivity analysis for shell structures, Int. J. Numer. Methods Engrg. 53 (9) (2002) 2087–2116.

[213] N.H. Kim, K.K. Choi, M.E. Botkin, Numerical method for shape optimization using meshfree method, Struct. Optim. 24 (6) (2003) 418–429.

[214] G. Bugeda, L. Gil, E. Onate, Structural shape sensitivity analysis for nonlinear material models with strain softening, Struct. Optim. 17 (1999) 162–171.

[215] C.A. Vidal, R.B. Haber, Design sensitivities for rate-independent elastoplasticity, Comput. Methods Appl. Mech. Engrg. 107 (1993) 393–431.

[216] K.K. Choi, S. Cho, Design sensitivity analysis for structures using explicit time integration, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2–4 September 1998, St Louis, MO.

[217] M. Kleiber, T.D. Hien, Parameter sensitivity of inelastic buckling and post-buckling response, Comput. Methods Appl. Mech. Engrg. 145 (1997) 239–262.

[218] H. Noguchi, T. Hisada, Sensitivity analysis in post-buckling problems of shell structures, Comput. Struct. 47 (4/5) (1993) 699–710.

[219] J.W. Hou, C. Mei, Y.X. Xue, Design sensitivity analysis of beams under nonlinear forced vibrations, AIAA J. 28 (6) (1990) 1067–1068.

[220] M.M. Godse, E.J. Haug, K.K. Choi, A parametric design methodology for concurrent engineering, Technical report, Center for Computer-Aided Design, The University of Iowa, Iowa City, IA, USA, 1991.

[221] R.J. Yang, M.J. Fiedler, Design modeling for large-scale three-dimensional shape optimization problems, in: Proceedings of the 1987 ASME International Computers in Engineering Conference and Exhibition, New York, USA, 1987, pp. 177–182.

[222] S. Kodiyalam, V. Kumar, P.M. Finnigan, A constructive solid geometry approach to three-dimensional shape optimization, AIAA J. 30 (5) (1992) 1408–1415.

[223] M.E. Botkin, Shape optimization of plate and shell structures, AIAA J. 20 (2) (1982) 268–273.

[224] C. Faure, C. Ducal, Shape optimal design—a performing cad oriented formulation, in: 25th AIAA SDM Conference, 1984, Palm Springs, CA, USA.

[225] R.J. Yang, M.E. Botkin, A modular approach for three-dimensional shape optimization, AIAA J. 25 (3) (1987) 492–497.

[226] M.H. Imam, Three-dimensional shape optimization, Int. J. Numer. Methods Engrg. 18 (1982) 661–673.

[227] K.K. Choi, Shape design sensitivity analysis and optimal design of structural systems, in: C.A.M. Soares (Ed.), Computed Aided Optimal Design, Springer-Verlag, Heidelberg, 1987, pp. 439–492.

[228] K.K. Choi, T.M. Yao, 3-d shape modeling and automatic regridding in shape design sensitivity analysis, in: Sensitivity Analysis in Engineering, NASA Conference Publication 2457, 1987, pp. 329–345.

[229] T.M. Yao, K.K. Choi, 3-d shape optimal design and automatic finite element regridding, Int. J. Numer. Methods Engrg. 28 (1989) 369–384.

[230] H.-G. Seoung, K.K. Choi, Boundary-layer approach to shape design sensitivity analysis, Mech. Struct. Mach. 15 (2) (1987) 241–263.

[231] J.C. Robinson, J.S. Campbell, D. Kelliher, Interior point tracking in shape evolving unstructured finite element meshes, Engrg. Comput. 15 (6) (1998) 721–731.

[232] S.D. Rajan, A.D. Belegundu, Shape optimal-design using fictitious loads, AIAA J. 27 (1) (1989) 102–107.

[233] A.D. Belegundu, S.D. Rajan, A shape optimization approach based on natural design variables and shape functions, Comput. Methods Appl. Mech. Engrg. 66 (1988) 89–106.

[234] G. Chiandussi, G. Bugeda, G. Onate, Shape variable definition with C^0, C^1 and C^2 continuity functions, Comput. Methods Appl. Mech. Engrg. 188 (2000) 727–742.

[235] K.H. Chang, K.K. Choi, A geometry-based parameterization method for shape design of elastic solids, Mech. Struct. Mach. 20 (2) (1992) 215–252.

[236] K.K. Choi, K.-H. Chang, A study of design velocity field computation for shape optimal design, Finite Element Anal. Des. 15 (1994) 317–341.

[237] P. Beckers, Recent developments in shape sensitivity analysis: the physical approach, Engrg. Optim. 18 (1991) 67–78.

[238] R.T. Haftka, R.V. Grandhi, Structural shape optimization—a survey, Comput. Methods Appl. Mech. Engrg. 57 (1986) 91–106.

[239] Y. Ding, Shape optimization of structures—a literature survey, Comput. Struct. 24 (6) (1986) 985–1004.