Struct Multidisc Optim (2007) 34: 61–73
DOI 10.1007/s00158-006-0061-7

RESEARCH PAPER

Nicholas M. K. Poon · Joaquim R. R. A. Martins

An adaptive approach to constraint aggregation using adjoint sensitivity analysis

Received: 22 December 2005 / Revised manuscript received: 13 June 2006 / Published online: 12 December 2006

Abstract Constraint aggregation is the key for efficient structural optimization when using a gradient-based optimizer and an adjoint method for sensitivity analysis. We explore different methods of constraint aggregation for numerical optimization. We analyze existing approaches, such as considering all constraints individually, taking the maximum of the constraints and using the Kreisselmeier–Steinhauser (KS) function. A new adaptive approach based on the KS function is proposed that updates the aggregation parameter by taking into account the constraint sensitivity. This adaptive approach is shown to significantly increase the accuracy of the results without additional computational cost especially when a large number of constraints are active at the optimum. The characteristics of each aggregation method and the performance of the proposed adaptive approach are shown by solving a wing structure weight minimization problem.

Keywords Constraint aggregation · Constraint handling · Adjoint sensitivity analysis · Kreisselmeier–Steinhauser function · Adaptive constraint aggregation

1 Introduction

For most engineering problems, design constraints are necessary and must be taken into account during the optimization. When considering large numbers of constraints, however, the computational cost of numerical optimization increases, and the convergence behavior is affected. In large-scale problems such as the multidisciplinary design optimization of aircraft, approaches are needed to minimize these effects to obtain the optimum in reasonable time with adequate accuracy. For problems with large numbers of design variables, only gradient-based algorithms, such as sequential quadratic programming (SQP), that utilize both function value and function gradient exhibit an acceptable convergence rate. When using gradient-based algorithms, the calculation of the objective function and constraint gradients can be the most time-consuming step in the optimization cycle depending on which method is used to estimate the gradient. The computation of the gradient can be performed efficiently by using algorithms such as the direct method, which is efficient for cases with few variables and many constraints. Alternatively, when computing the gradients of a few functions with respect to many variables, an adjoint method is better suited. The adjoint method was first used to perform sensitivity analysis of structures by Haug and Feng (1978) and has since been applied to numerous structural optimization problems (Haftka and Gürdal 1993), including the aerostructural design of aircraft configurations using a coupled-adjoint approach (Martins et al. 2004).

For problems with large numbers of both design variables and constraints, however, there are no methods that can compute the full sensitivity matrix efficiently. By using constraint aggregation, constraints are lumped into a single composite function. With only a single or few composite constraints, the adjoint method can compute the sensitivities efficiently. In addition, constraint aggregation reduces discontinuities at the intersection of constraints by forming a continuously differentiable function.

The Kreisselmeier–Steinhauser (KS) function is a widely used constraint aggregation method for gradient-based optimization, which has been used in various applications, including aerodynamic shape optimization (Anderson and Bonhaus 1997), chemical process design (Rooney and Biegler 2002), and aircraft design (Stettner and Schrage 1992; Akgün et al. 2001; Martins et al. 2004).

Our main interest in constraint aggregation stems from its application to a high-fidelity, aerostructural optimization framework that optimizes aircraft configurations with respect to both the aerodynamics and structures (Alonso et al. 2004). The structural module of this framework uses the KS function to aggregate the stress constraints so as to take advantage of the coupled-adjoint method that it employs to perform efficient aerostructural sensitivity analysis (Martins et al. 2005).

N. M. K. Poon · J. R. R. A. Martins (✉)
University of Toronto Institute for Aerospace Studies,
4925 Dufferin St., Toronto, ON M3H5T6, Canada
e-mail: mkpoon@utoronto.ca
e-mail: martins@utias.utoronto.ca

The objective of this article is twofold. First, we would like to study the properties of the KS function and examine the solutions it yields. In the current literature, there is no in-depth analysis of the use of the KS function in numerical optimization. From previous applications of the KS function (Haftka and Gürdal 1993; Akgün et al. 2001), it is well known that it yields conservative results. However, the optima and the convergence behavior were not compared to a reference case where constraints are considered separately.

Second, we propose an adaptive constraint aggregation method based on the original KS function that yields more accurate optimization results. The improvements are illustrated by the weight minimization of a wing structure, where the new method is compared with the original KS function, the maximum constraint approach, and the reference case, for which the constraints are considered separately.

In the following section, we present the different constraint aggregation methods and discuss the theory behind them. We then proceed to present and discuss results from the structural weight minimization of an aircraft wing.

2 Motivation

Within our aircraft design framework, the main utility of the KS function is to make the coupled-adjoint sensitivity calculation of stress constraints more efficient. While our goal is to closely examine the optimization with adjoint sensitivities of the composite constraint, only the structural optimization will be studied. Consider the typical structural optimization problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & W(\mathbf{x}) \\ \text{subject to} \quad & \frac{\boldsymbol{\sigma}(\mathbf{x})}{\boldsymbol{\sigma}_{\text{yield}}} - 1 \leq 0, \quad \boldsymbol{\sigma} \in \mathbb{R}^m \\ & \mathbf{x}_{\min} - \mathbf{x} \leq 0. \end{aligned} \tag{1}$$

where W is the structural weight, $\mathbf{x}$ is the vector of design variables, $\boldsymbol{\sigma}$ represents the stresses in each finite element of the structural model, and σ_{yield} is the yield stress of the material. When using high-fidelity models, there may be $\mathcal{O}(10^6)$ elements and the number of constraints may be of the same order or greater, especially if buckling constraints are also considered.

When the sensitivities of the constraint functions are calculated using finite differences, the number of required function evaluations is proportional to the number of design variables. Therefore, finite differencing is only feasible for cases with few design variables or when the cost of the analysis is very low.

The adjoint method, however, is a method whose computational cost is independent of the number of design variables. The governing equations are only solved once, followed by the adjoint calculation of the sensitivities. The state variables of the structural analysis are the displacements, $\mathbf{u}$, which are obtained by solving the governing equation

$$\mathbf{R} = \mathbf{K}\mathbf{u} - \mathbf{F} = 0, \tag{2}$$

where $\mathbf{K}$ is the stiffness matrix, and $\mathbf{F}$ represents the applied forces.

Applying the chain rule to the function of interest, I, we obtain its total sensitivity with respect to the design variables

$$\frac{\mathrm{d}I}{\mathrm{d}\mathbf{x}} = \frac{\partial I}{\partial \mathbf{x}} + \frac{\partial I}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}}. \tag{3}$$

The functions of interest in our structural optimization problem (1) are the weight and the stresses. However, unlike the stresses, the weight does not depend on the displacements and therefore does not require a sensitivity analysis method that accounts for the governing equations.

We also know that the total sensitivity of the governing equation residuals must be zero and, therefore,

$$\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} = 0. \tag{4}$$

Solving for the sensitivity of the displacements with respect to the design variables, we obtain

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} = -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}}, \tag{5}$$

and substituting this results into the total sensitivity (3) yields

$$\frac{\mathrm{d}I}{\mathrm{d}\mathbf{x}} = \frac{\partial I}{\partial \mathbf{x}} - \underbrace{\frac{\partial I}{\partial \mathbf{u}}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right]^{-1}}_{\boldsymbol{\psi}}\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{6}$$

From this equation, we can see that the matrix of sensitivities $\partial\mathbf{R}/\partial\mathbf{u}$ can be factorized with one of two possible right-hand sides: $-\partial I/\partial\mathbf{u}$ or $\partial\mathbf{R}/\partial\mathbf{x}$. Factorizing the matrix with a column of the latter term is the basis for the direct method. Factorizing with the former vector, one eliminates the dependence on the design variables and obtains the adjoint equation

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\boldsymbol{\psi} = -\frac{\partial I}{\partial \mathbf{u}}. \tag{7}$$

For each function of interest, a different adjoint vector is required. As a result, the cost of the adjoint method is proportional to the number of functions of interest, and it is most efficient when the system has one or few outputs. Thus, the application of the adjoint method to our optimization problem (1) is not advantageous when the number of finite elements is large.

There are cases for which a factorization of $\partial\mathbf{R}/\partial\mathbf{u}$ can be performed once and stored in memory. For these cases, solving the adjoint (7) with multiple right-hand sides can be a very efficient procedure. However, in the aerostructural optimization framework mentioned previously, the adjoint calculation is coupled, involving computational fluid dynamics, and explicit computation of the factorization is impractical.

This invariably results in an expensive procedure when computing the sensitivities of many functions.

By assuming that the constraints are continuous—but not necessarily continuously differentiable—the constraints can be aggregated into a composite constraint. Through this reformulation, multiple constraints are aggregated into the KS function, and the modified optimization problem can be written as

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad W(\mathbf{x}) \\ &\text{subject to} \quad KS[\boldsymbol{\sigma}(x)] \le 0, \quad \boldsymbol{\sigma} \in \mathbb{R}^m \\ &\qquad\qquad \mathbf{x}_{\min} - \mathbf{x} \le 0. \end{aligned} \tag{8}$$

In this modified optimization problem, the KS function is a scalar that represents a conservative estimate of the maximum of all the constraints at any point in the design space. We will present the KS function in more detail in the following section.

3 Theory

In this section, we define the original optimization problem, describe the two methods that are currently used for constraint aggregation, and present the method that we propose.

3.1 Standard approach: individual constraints

This approach consists in solving the original problem formulation, which considers all constraints separately. This can be written as

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le 0, \quad \mathbf{g} \in \mathbb{R}^m. \end{aligned} \tag{9}$$

3.2 The maximum constraint approach

The maximum constraint approach is the simplest constraint aggregation method. The most violated constraint is considered, while the remaining constraints are ignored. The optimization problem for this approach is defined as

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\mathbf{x}) \\ &\text{subject to} \max\left[\mathbf{g}(\mathbf{x})\right] \le 0, \quad \mathbf{g} \in \mathbb{R}^m. \end{aligned} \tag{10}$$

A severe problem with this approach is that during the optimization process, the search direction is determined by considering only the most violated constraint, usually leading to the violation of another constraint in the next iteration. Most optimization algorithms find it extremely difficult to solve an optimization problem of this form.

3.3 Constraint aggregation using the Kreisselmeier–Steinhauser function

The KS function was first presented by Kreisselmeier and Steinhauser (1979). The function uses a "draw-down" factor or aggregation parameter, ρ, which is analogous to the penalty factor in penalty methods sometimes used in constrained optimization.

The function produces an envelope surface that is C_1 continuous and represents a conservative estimate of the maximum among the set of functions (Wrenn 1989). This formulation was first used to aggregate multiple objectives and constraints into single functions and was widely used before direct constrained optimization techniques became popular.

The KS function can be used to aggregate just constraints into single composite function defined as

$$KS\left[\mathbf{g}(\mathbf{x})\right] = \frac{1}{\rho} \ln\left[\sum_{j=1}^{m} e^{\rho g_j(\mathbf{x})}\right], \tag{11}$$

This definition is mostly used with conjunction with the SQP and trust-region methods (Akgün et al. 1999, 2001; Martins et al. 2004).

The properties of the KS function as derived by Raspanti et al. (2000) are listed below.

1. $KS(\mathbf{x}, \rho) \ge \max\left[\mathbf{g}(\mathbf{x})\right]$ for all $\rho > 0$.
2. $\lim_{\rho\to\infty} KS(\mathbf{x}, \rho) = \max\left[\mathbf{g}(\mathbf{x})\right]$.
3. $KS(\mathbf{x}, \rho_2) \ge KS(\mathbf{x}, \rho_1)$ for all $\rho_2 > \rho_1 > 0$.
4. $KS(\mathbf{x}, \rho)$ is convex if and only if all constraints are convex.

These properties are vital for the successful use of the KS function as a constraint aggregation technique in nonlinear programming. Properties 1 and 2 indicate that the KS function overestimates the maximum of the constraints, and therefore a positive value of KS is returned if a constraint is violated or close to being violated. Property 3 relates the conservativeness of the estimate in violation to the magnitude of the aggregation parameter, ρ. As ρ increases, the KS function approaches the maximum constraint at the current design point. Property 4 implies that the use of the KS function does not alter the convexity of a problem. Therefore, if the original problem is convex, the modified problem with constraints aggregated into KS function remains convex.

To reduce numerical difficulties caused by numerical overflow, we use an alternate formulation of the KS function given by

$$KS\left[\mathbf{g}(\mathbf{x})\right] = g_{\max}(\mathbf{x}) + \frac{1}{\rho} \ln\left[\sum_{j=1}^{m} e^{\rho\left(g_j(\mathbf{x}) - g_{\max}(\mathbf{x})\right)}\right], \tag{12}$$

where $g_{\max}$ is the maximum of all constraints evaluated at the current design point, $\mathbf{x}$.

The magnitude of the KS function value at a particular design point is bounded from above and below as follows:

$$g_{\max}(\mathbf{x}) < KS\left[\mathbf{g}(\mathbf{x})\right] < g_{\max}(\mathbf{x}) + \frac{\ln m}{\rho}. \quad (13)$$

The lower bound (most accurate) value is $g_{\max}(\mathbf{x})$. The upper bound (least accurate) value is inversely proportional to the "draw-down" factor, ρ, that determines the difference between the KS function and the maximum value of the constraint. As ρ approaches infinity, the KS function becomes equivalent to $g_{\max}$, the maximum of the all the constraints. The sum of exponentials and the factor ρ ensure that the value of the KS function is dominated by the largest values in $\mathbf{g}$. When a given constraint is not active, its contribution is several orders of magnitude smaller than that of active constraints.

From (13), we can calculate the maximum error for a particular value of ρ or the ρ value from a chosen maximum error. A machine-zero error could theoretically be achieved by choosing a large enough ρ. However, obtaining numerical estimates of second derivatives at the design points where the constraints intersect poses difficulties because the curvature at those points can be very large. Estimating the Hessian of the KS function at such points yields an ill-conditioned matrix when ρ is too large, which causes numerical difficulties. Therefore, $\rho = 50$ is a reasonable value that is often used (Akgün et al. 2001; Raspanti et al. 2000; Wrenn 1989).

In the following single-variable example, the application of the KS function as a constraint aggregation method and the effect of increasing ρ for inequality constraints can be visualized. Consider the following convex inequality constraints:

$$g_1(x) = \frac{5}{\ln(x)} - \frac{x}{5} - 4 \leq 0 \quad (14)$$

$$g_2(x) = \frac{x^2}{40} + \frac{x}{5} - 2 \leq 0 \quad (15)$$

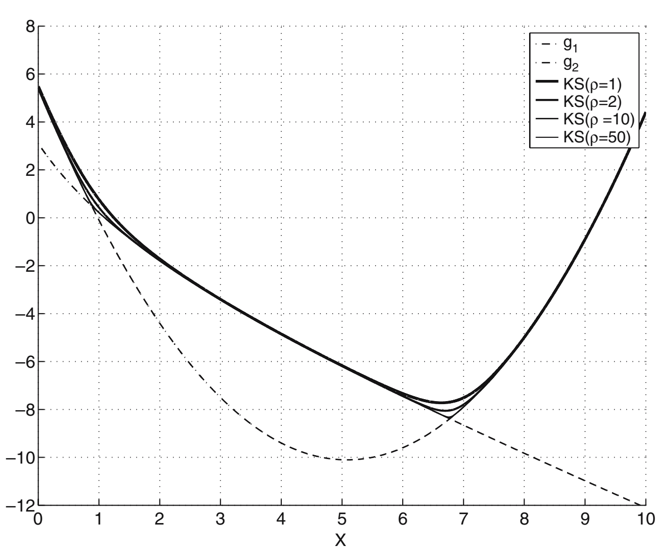


Fig. 1 KS function of two inequality constraints for increasing ρ

The KS function of these two constraints forms the envelope shown in Fig. 1. Note that the convexity of the KS function is inherited from the original constraints, and therefore the KS function is also convex. Also, as ρ increases, the KS function approaches the maximum constraint, $g_{\max}$. This is most noticeable at the intersection of the two constraints.

3.4 Adaptive approach for constraint aggregation

The original KS approach can be classified as nonadaptive as it carries the predetermined aggregation parameter ρ throughout the optimization. Although this parameter (typically 50 or less) is selected to avoid numerical problems, the function returns a conservative value, and therefore this widely adopted approach results in inaccuracies in the optimization results. These inaccuracies increase with the number of active constraints, as shown in Fig. 2 for an analytic optimization problem (Qin and Nguyen 1994). This problem is selected because it is representative of the behavior at the vicinity of local optima, which is typical of structural optimization problems with stress constraints.

The error in the optimum is 10% for 100 constraints and increases to 15% for 2,500 constraints. In practical finite element models, for which $\mathcal{O}(10^4)$ elements—and thus the same order of stress constraints—are routinely used, the error would exceed %18, which is unacceptable in most cases. Note that the relationship shown in Fig. 2 agrees with the maximum error determined from (13).

This suboptimal result suggests that the traditional approach poses difficulties to the optimizer, which prevent it for converging to the true optimum. The main reason for this error is the fact that the feasible region defined by the KS function does not necessarily contain the true optimum. As shown in Fig. 3, for an example, with one design variable and two inequality constraints, where $f(x)$ is the objective, and $g_1(x)$, $g_2(x)$ are the constraints, the objective function changes more rapidly across the domain than the constraints. It is clear that the KS function in this case defines a smaller feasible region than the true feasible region defined by the

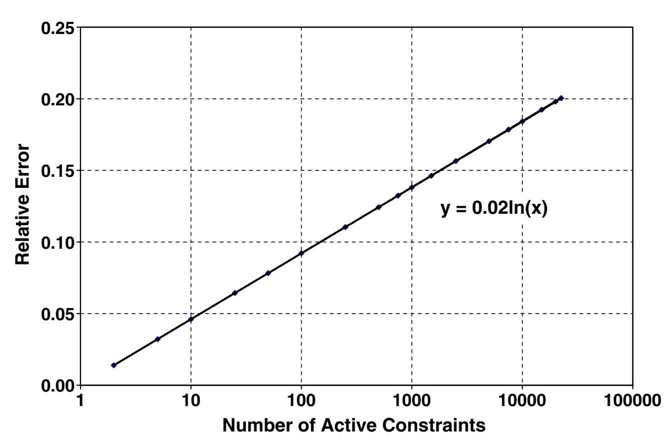


Fig. 2 Relative error in optimization result using the KS function with $\rho = 50$

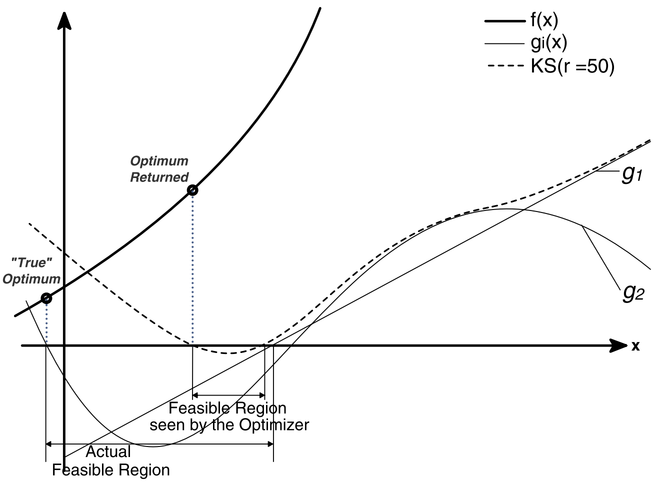


Fig. 3 Feasible region reduction when using the KS function

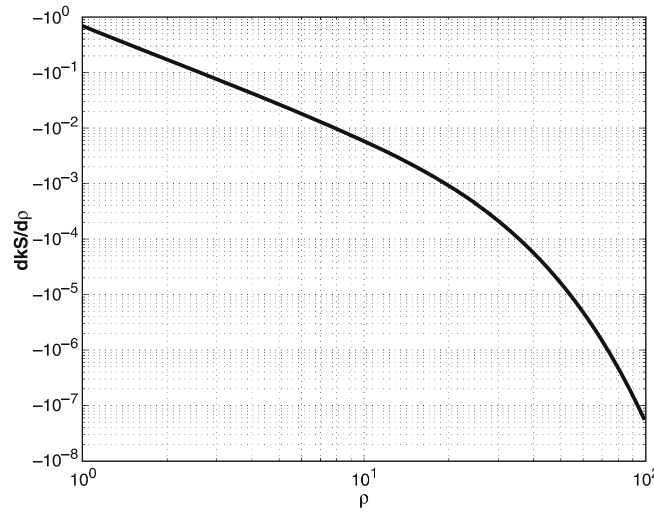


Fig. 5 KS' vs ρ near constraint intersection

original constraints. Thus, the actual feasible region is not visible to the optimizer if the KS function is used. This effect becomes more significant when the optimum is located at a point where multiple constraints are active.

As previously mentioned, although the KS function becomes a better approximation for increasing values of ρ, this causes numerical difficulties due to an ill-conditioned Hessian. Therefore, the value of the aggregation parameter should only be increased as required. This can be achieved by defining a nominal aggregation parameter at the beginning of the optimization, and by increasing it as needed according to the sensitivity of KS with respect to the aggregation parameter, $KS' = \mathrm{d}KS/\mathrm{d}\rho$, at the current design point.

At design points away from constraints' intersections, KS' is zero as there is either no active constraint or only one. At design points close to the intersection of constraints, this becomes nonzero and reaches a maximum when the design point is at the constraints' intersection, as dictated by (13).

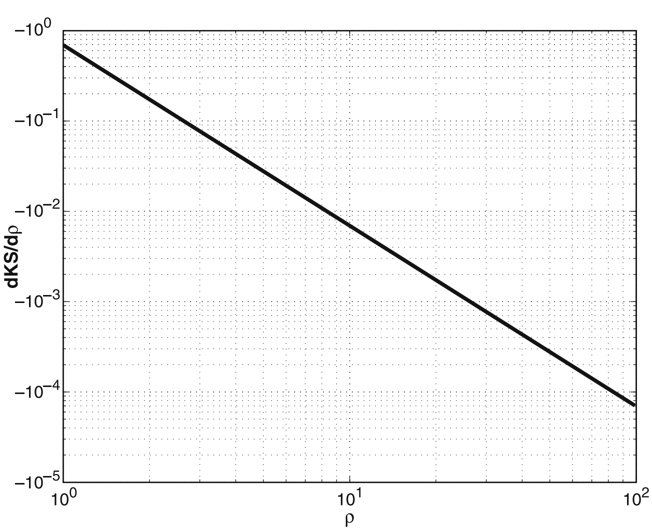


Fig. 4 KS' vs ρ at constraint intersection

Figures 4 and 5 show the KS sensitivity for increasing ρ for design points at and nearby the constraints' intersection.

In the adaptive KS approach, the aggregation parameter is increased such that KS' is less than or equal to an acceptable tolerance, say, 10^{-6}. This is the desired value, KS'_d. Given the desired value and the current value, KS'_c, we may derive a relationship by considering the slope in the logarithmic scale

$$\frac{\log KS'_1 - \log KS'_c}{\log \rho_1 - \log \rho_c} = \frac{\log KS'_d - \log KS'_c}{\log \rho_d - \log \rho_c} \tag{16}$$

This relationship is derived from Fig. 4, and it is essentially the secant method applied to $\log KS'$.

The subscript 1 represents a value calculated at a finite step from the current point. In our application, $\rho_1 = \rho_c + 10^{-3}$. By rearranging (16), the desired value, ρ_d, can be expressed as

$$\log \rho_d = \log\left(\frac{KS'_d}{KS'_c}\right)\left[\log\left(\frac{KS'_1}{KS'_c}\right)\right]^{-1} \log\left(\frac{\rho_1}{\rho_c}\right) + \log \rho_c \tag{17}$$

In our implementation, ρ_c is set to 50. During the optimization process, ρ_c is kept constant and ρ_d is used for the calculation of KS and $\mathrm{d}KS/\mathrm{d}x$ if KS'_c does not meet the prescribed tolerance. Also, KS' is calculated using complex-step derivative approximation to obtain accurate results (Martins et al. 2003). The resulting algorithm is described below.

The following information is given:

– Constraint functions: $\mathbf{g}(\mathbf{x})$, $j = 1, \ldots, m$
– Initial aggregation parameter, ρ_c
– Desired KS sensitivity, KS'_d

When a constraint evaluation is requested by the optimizer, the following sequence is executed:

1. Compute the KS function using the current parameters.
2. Compute KS'_c at current design point using ρ_c and the constraint values.

3. If $KS'_c < KS'_d$, return the current KS value; otherwise, continue.
4. Compute ρ_d using (17).
5. Compute KS using ρ_d and return its value.

In the next section, we demonstrate that this algorithm addresses the shortcomings of the conventional KS function approach in a practical structural optimization problem.

4 Results

To demonstrate the application of the adaptive approach, a weight optimization study is conducted for a wing structure modeled with multiple spars and ribs, with a total of 45 tube elements as illustrated in Fig. 6. The finite element model uses a modified three-dimensional frame elements that account for one translational degree of freedom (vertical) and two rotational degrees of freedom (axial and transversal) at each node, thus each element has a total of six degrees of freedom.

The design variables are the diameters of each tube element, the objective function is the weight of the structure, and the constraints are the stresses in each element. The optimization problem can be stated as follows:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^{45}}{\text{minimize}} \quad W(\mathbf{x}) \\ &\text{subject to} \quad \frac{\boldsymbol{\sigma}(\mathbf{x})}{\boldsymbol{\sigma}_{\text{yield}}} - 1 \le 0, \quad \boldsymbol{\sigma} \in \mathbb{R}^{45} \\ &\qquad\qquad \mathbf{x}_{\min} - \mathbf{x} \le 0. \end{aligned} \tag{18}$$

Note that the largest fraction of the computational cost for this problem is the evaluation of the constraints, which requires the solution of the finite element equations and the calculation of the stresses. Also, the sensitivity computation of the constraints involves a large number of inputs (design variables) and a large number of outputs (stress of each element), and therefore constraint aggregation is required to use adjoint method efficiently.

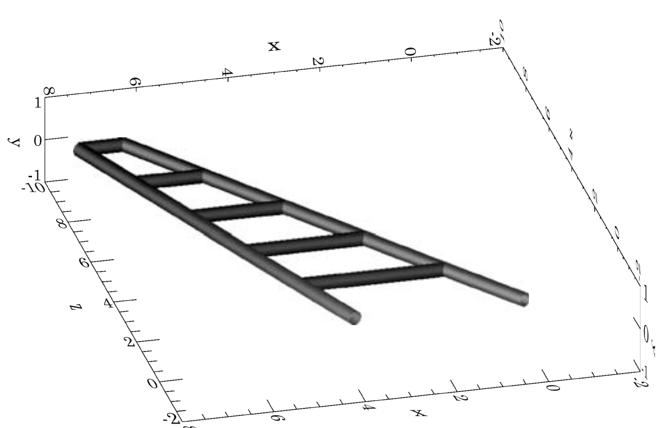


Fig. 6 Finite element model of the wing structure

Table 1 Computational cost of computing sensitivities for 45 design variables and 45 stress constraints, 1 U = 4 s

	IC	Max	KSC	AKS
Constraint vector evaluation	1	1	1	1
Forward difference	46	46	46	46
Central difference	91	91	91	91
Complex step	54	54	54	54
Adjoint method with forward-difference partial derivatives				
$\partial I/\partial \mathbf{x}$	0	1	1	1
$\partial I/\partial \mathbf{u}$	0	1	1	1
$\partial \mathbf{R}/\partial \mathbf{u}$	0	0	0	0
$\partial \mathbf{R}/\partial \mathbf{x}$	0	0	0	0
Adjoint solution, $\boldsymbol{\psi}$	44	1	1	1
Total	48	4	4	4
Adjoint method with central-difference partial derivatives				
$\partial I/\partial \mathbf{x}$	1	1	1	1
$\partial I/\partial \mathbf{u}$	1	1	1	1
$\partial \mathbf{R}/\partial \mathbf{u}$	0	0	0	0
$\partial \mathbf{R}/\partial \mathbf{x}$	1	1	1	1
Adjoint solution, $\boldsymbol{\psi}$	44	1	1	1
Total	49	5	5	5
Adjoint method using complex-step partial derivatives				
$\partial I/\partial \mathbf{x}$	1	1	1	1
$\partial I/\partial \mathbf{u}$	1	1	1	1
$\partial \mathbf{R}/\partial \mathbf{u}$	0	0	0	0
$\partial \mathbf{R}/\partial \mathbf{x}$	1	1	1	1
Adjoint solution, $\boldsymbol{\psi}$	45	1	1	1
Total	48	5	5	5

4.1 Optimization settings

Results for all constraint handling methods described in Section 3 were obtained. The reference method is the one that considers all individual constraints separately (IC). The constraint aggregation methods are the original KS function (KSC), the maximum function (Max), and the new adaptive KS function (AKS).

The adaptive constraint aggregation method was implemented in a Python-based application program interface that

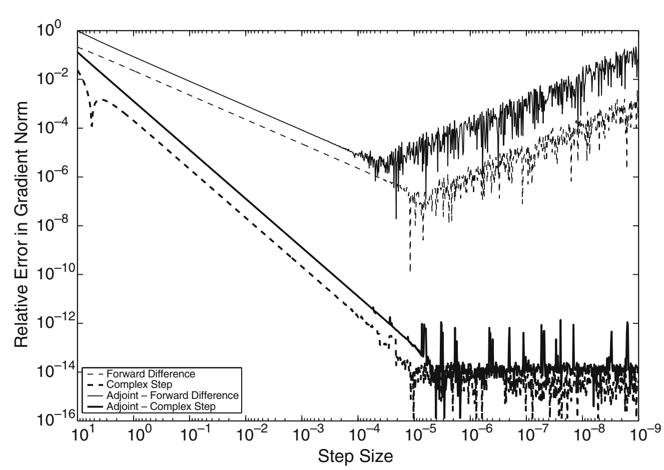


Fig. 7 Gradient relative error vs step size

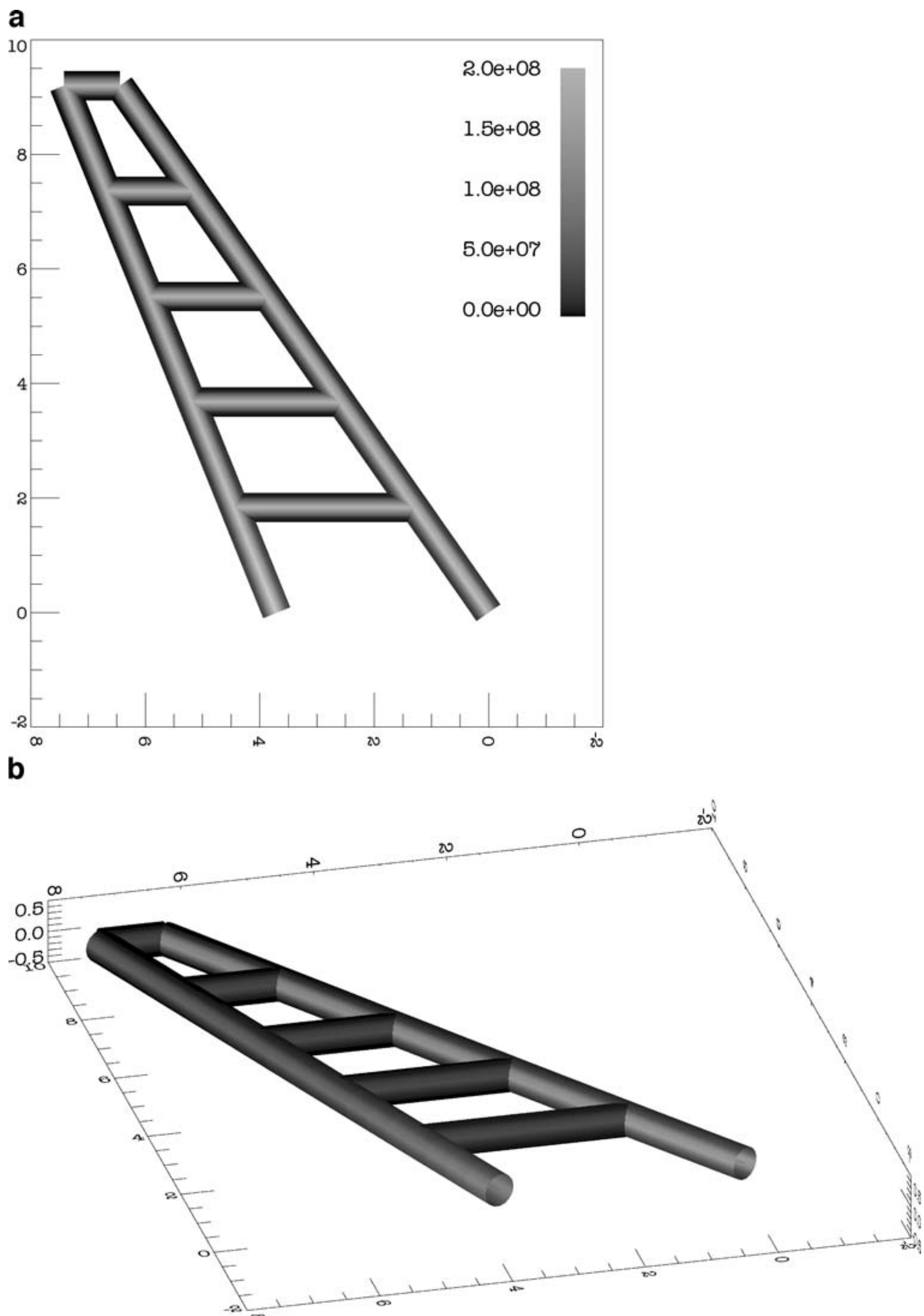


Fig. 8 Initial structure, mass = 2741.351 kg

connects the optimizer, SNOPT, and the finite element solver. SNOPT is an active-set SQP-based optimizer (Gill et al. 2002).

4.2 Sensitivity analysis

The computational costs for function and gradient evaluations of the various constraint handling methods are listed in Table 1. The first column (IC) corresponds to the computation of a 45 × 45 sensitivity matrix, while for the remaining columns, only a vector with 45 components needs to be computed. The constraint aggregation reduced the sensitivity computation cost of the adjoint method by one order of magnitude.

Because the accuracy of finite differences varies widely with step size, a study was conducted to establish the precision of the sensitivity results. Figure 7 shows the relative error

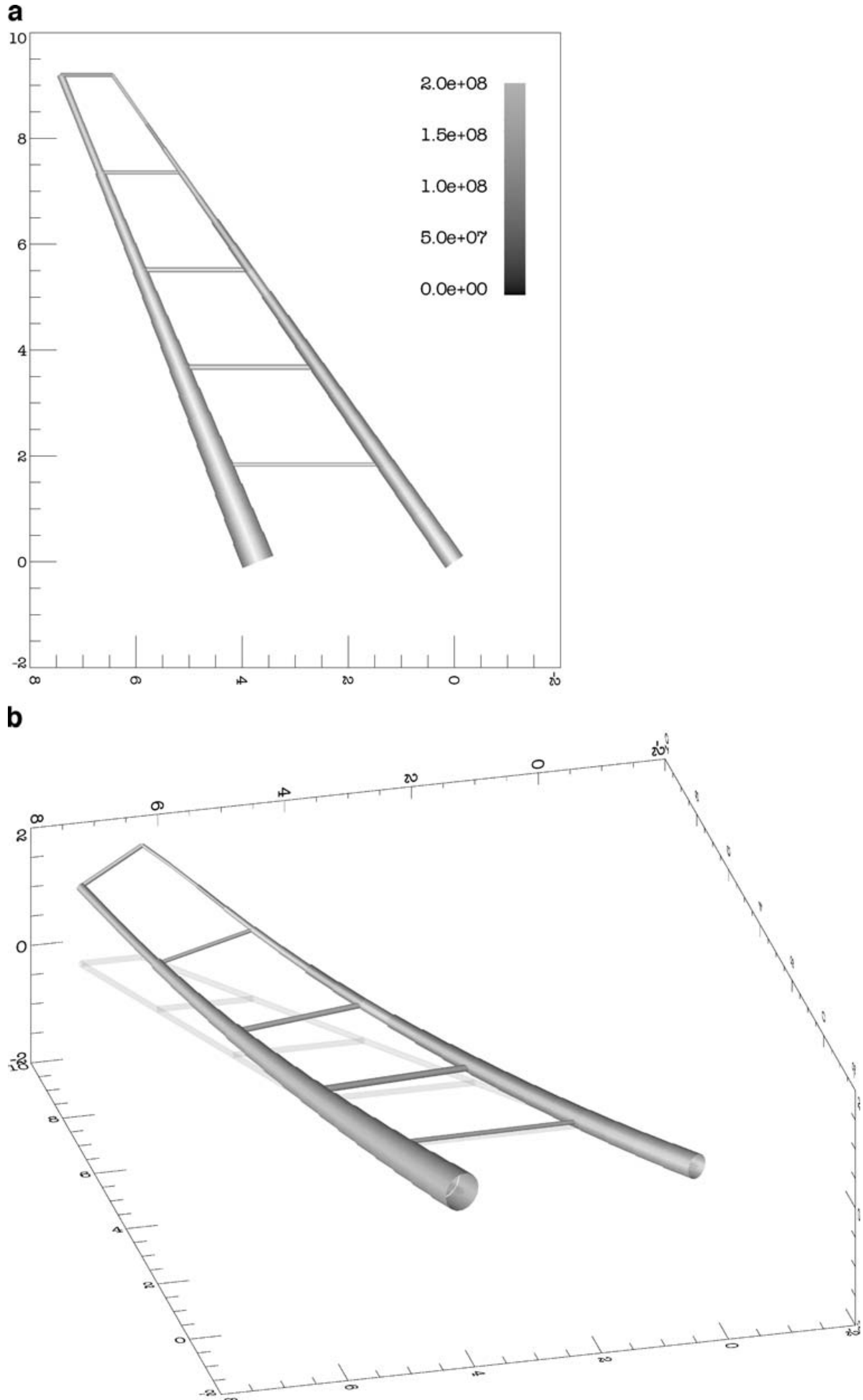
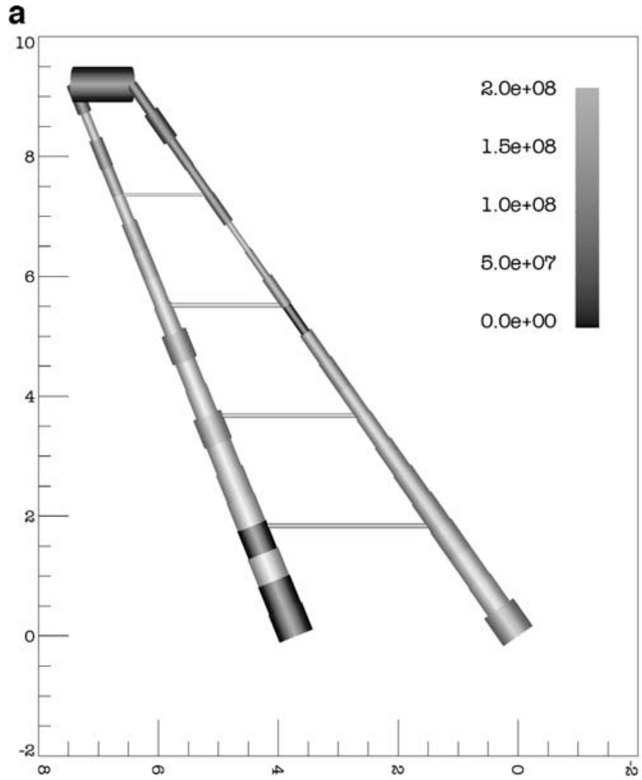
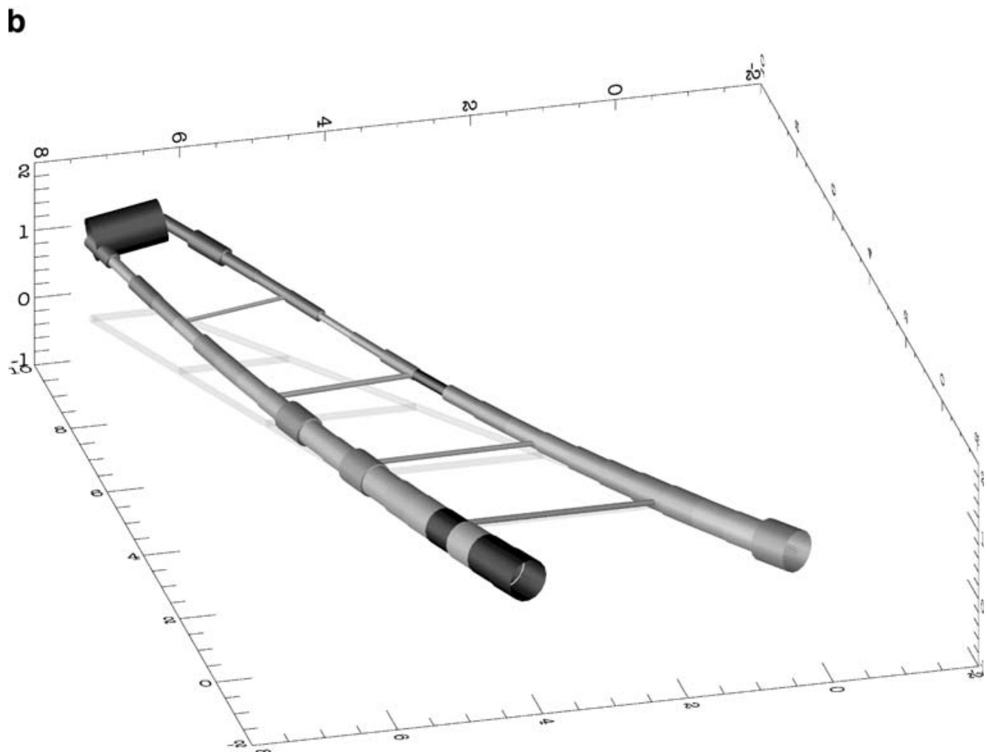


Fig. 9 Optimized structure with individual constraints (IC), mass = 183.500 kg

Fig. 10 Optimized structure using maximum of constraints (Max), mass = 247.152 kg





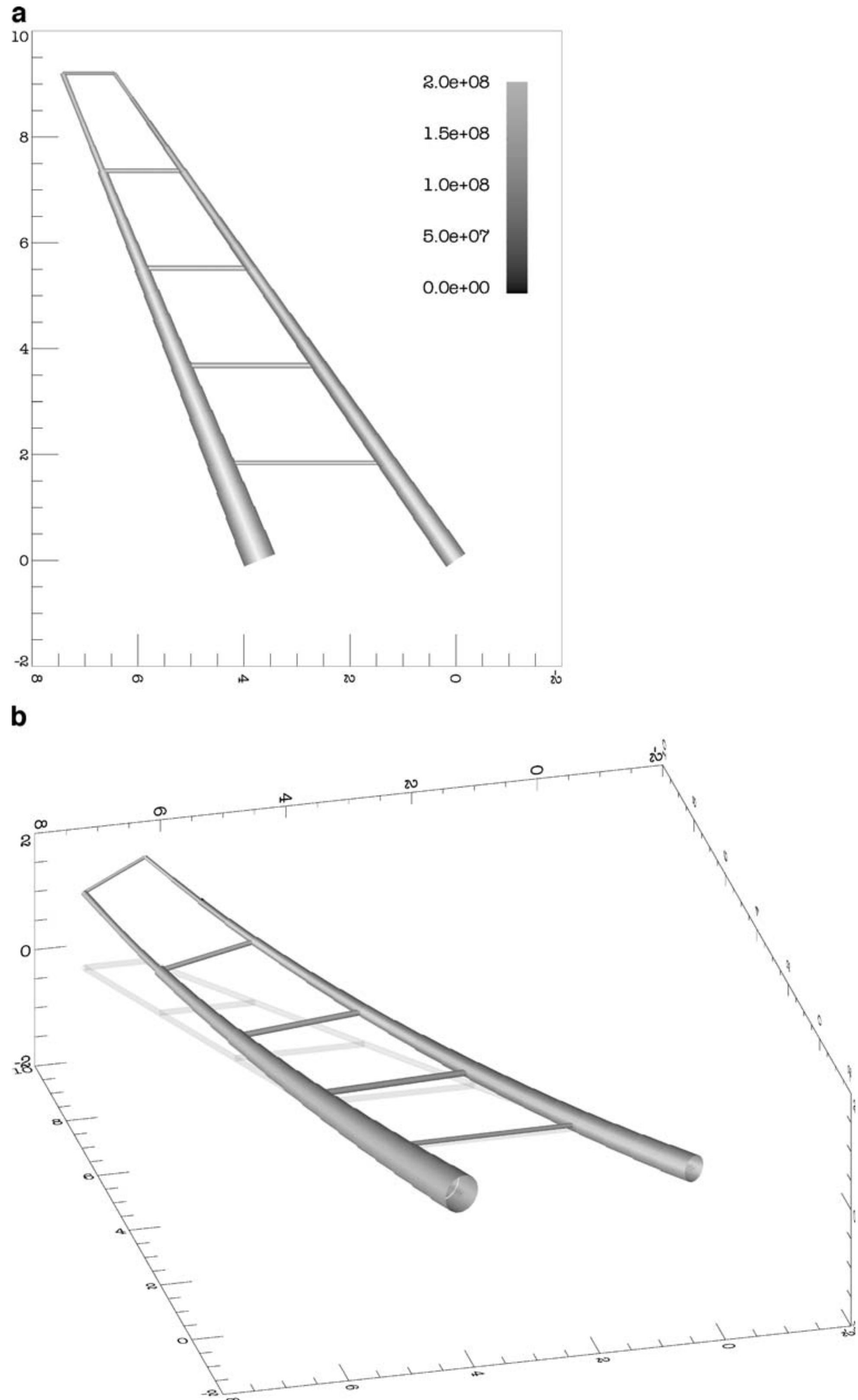


Fig. 11 Optimized structure using the KS function (KSC) with $\rho = 50$, mass = 194.224 kg

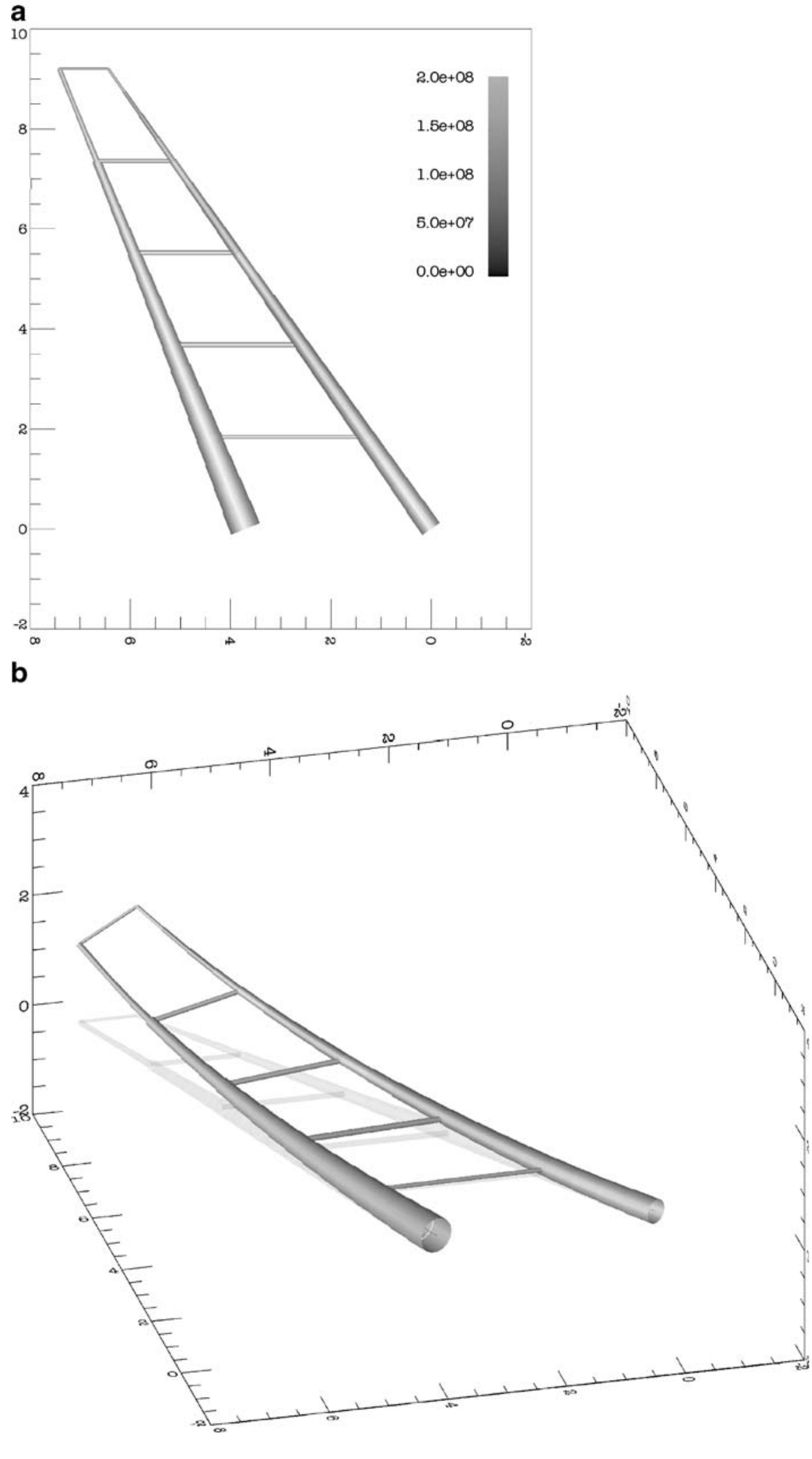


Fig. 12 Optimized structure using the adaptive KS function (AKS), mass = 183.076 kg

in the gradient norm of stresses with respect to an element thickness vs decreasing step size for four different sensitivity analysis methods.

In this figure, we can see that the forward difference estimates exhibit linear convergence with decreasing step size, which is consistent with the fact that the error term in the forward difference formula is $\mathcal{O}(h)$. However, as the step size decreases below 10^{-5}, the relative error increases due to subtractive cancellation.

The complex-step method exhibits quadratic convergence, which is consistent with the error term in the complex-step derivative approximation (Martins et al. 2003). Unlike the forward difference, however, the complex-step method is not subject to subtractive cancellation, and the relative error remains at machine zero for step sizes below 10^{-5}.

The adjoint method we used is semianalytic, i.e., the partial derivatives in the adjoint (6) were computed using either finite differences or the complex-step method. In Fig. 7, we can see that the adjoint result using forward differences is more accurate than the pure forward difference but is still subject to the same subtractive cancellation errors.

The adjoint method with partial derivatives computed using the complex-step method, on the other hand, exhibits much better accuracy with a relative error of 10^{-14} that is step-size insensitive for steps below 10^{-6}. These are the expected results that have been thoroughly analyzed by Martins et al. (2003). Thus, the adjoint method with partial derivatives computed using the complex-step method is used in the results that follow.

4.3 Optimization results

Figure 8 shows the initial structure with design variables arbitrarily set to unity. The reference optimization case that considers all constraints separately resulted in a minimum mass of 183.5 kg and a fully stressed structure, as shown in Fig. 9.

The optimization results using the different constraint aggregation methods are shown in Figs. 9, 10, 11, and 12. The result corresponding to the maximum of constraints formulation, shown in Fig. 10, is clearly infeasible because several constraints are violated.

The KS function formulation (Fig. 11) converged to a minimum, but the resulting structure was not quite fully stressed, and the final weight was 5.8% higher than the reference result. The adaptive KS approach achieved better accuracy, with an error of only 0.2%, as seen in Fig. 12.

4.4 Convergence history

The convergence history of the optimization is shown in Fig. 13 for all constraint handling methods. The logarithmic axis of the figure shows the relative difference between the value of the objective function and the reference minimum of the problem. This is plotted against the wall-clock time to show the convergence behavior and accuracy of the final result.

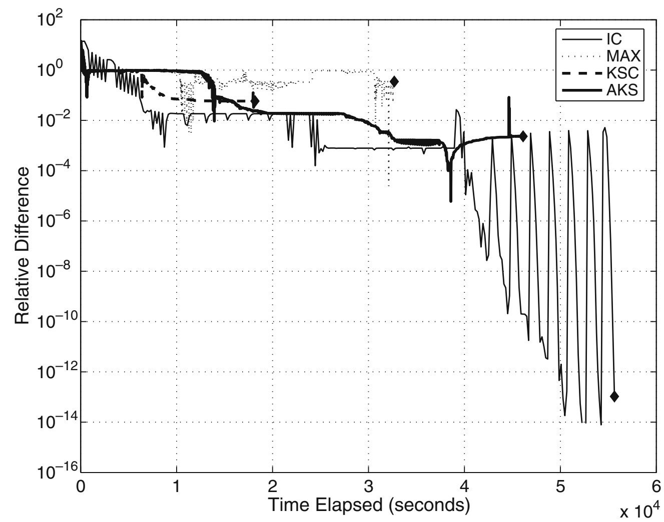


Fig. 13 Relative difference between the objective function and reference optimum vs wall-clock time

From this figure, we can see that the maximum constraint approach demonstrated poor convergence behavior. Using this approach, we could not obtain a converged solution. This is due to the lack of first order continuity in the constraint, which prevents the optimizer from properly estimating the Lagrange multiplier and thus leads to difficulties when using the active-set SQP optimizer.

The original KS function converges to a minimum in 30% of the time taken by the reference case and the result. Because the optimum is located at the intersection of active constraints, using a fixed aggregation parameter leads to reduction of the feasible region as seen by the optimizer and thus the optimum is not as accurate as the reference. Using

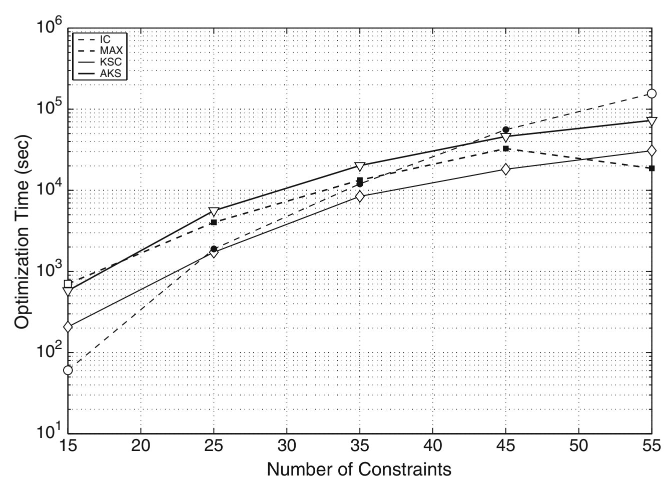


Fig. 14 Optimization performance vs number of constraints

the adaptive approach, the optimization converges in 75% of the reference time.

5 Optimization performance vs problem size

The sample optimization results demonstrated a better accuracy relative to the original KS function and reduced computational cost compared to treating the constraints individually. We now study the impact of increasing the number of constraints on the performance of the optimization.

Figure 14 shows the total optimization time vs the number of constraints. The white colored symbols represent feasible optimum, and the black symbols represent cases that did not converge. Optimization with both the original KS formulation and the adaptive approach returns feasible optima for all cases that we ran. Although the adaptive KS approach incurs a higher computational cost, it is much more accurate and thus is a better choice when precision is important.

6 Conclusion

The motivation, derivation, and application of an adaptive constraint aggregation method were presented. A widely used constraint aggregation method, the KS function, utilizes the full potential of adjoint sensitivities by reducing the number of constraints. This is efficient for large-scale problems with high computational cost for each function evaluation. The original KS formulation has a conservative nature that leads to suboptimal results especially when many constraints are active at the optimum. The adaptive approach proposed in the present work avoids this problem by updating the aggregation parameter according to the constraint sensitivity. This leads to a guaranteed accuracy in the result and is more robust than using large magnitude of the aggregation parameter in the original formulation.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council and the Canada Research Chairs program.

References

Akgün MA, Haftka RT, Wu KC, Walsh JL (1999) Sensitivity of lumped constraints using the adjoint method. In: Proceedings of the 40th structures, structural dynamics and materials conference, St. Louis, Missouri AIAA Paper 99–1314

Akgün MA, Haftka RT, Wu KC, Walsh JL, Garcelon JH (2001) Efficient structural optimization for multiple load cases using adjoint sensitivities. AIAA J 39(3):511–516

Alonso JJ, LeGresley P, van der Weide E, Martins JRRA, Reuther JJ (2004) pyMDO: a framework for high-fidelity multi-disciplinary optimization. AIAA Paper 2004–4480

Anderson WK, Bonhaus DL (1997) Aerodynamic design on unstructured grids for turbulent. NASA TM-112867, NASA Langley Research Center, Hampton, Virginia 23681-0001

Gill PE, Murray W, Saunders MA (2002) SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM J Optim 12(4):979–1006

Haftka RT, Gürdal Z (1993) Elements of structural optimization, 3rd edn. Kluwer, Boston, MA

Haug EJ, Feng TT (1978) Optimal design of dynamically loaded continuous structures. Int J Numer Methods Eng 12(2):299–317

Kreisselmeier G, Steinhauser R (1979) Systematic control design by optimizing a vector performance index. In: International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems, Zurich, Switzerland

Martins JRRA, Sturdza P, Alonso JJ (2003) The complex-step derivative approximation. ACM Trans Math Softw 29(3):245–262. http://doi.acm.org/10.1145/838250.838251

Martins JRRA, Alonso JJ, Reuther JJ (2004) High-fidelity aerostructural design optimization of a supersonic business jet. J Aircr 41(3):523–530

Martins JRRA, Alonso JJ, Reuther JJ (2005) A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. Optim Eng 6(1):33–62. http://www.kluweronline.com/issn/1389-4420/contents

Qin J, Nguyen DT (1994) Generalized exponential penalty function for nonlinear programming. Comput Struct 50(4):509–513

Raspanti C, Bandoni J, Biegler L (2000) New strategies for flexibility analysis and design under uncertainty. Comput Chem Eng 24(2000):2193–2209

Rooney WC, Biegler LT (2002) Optimal process design with model parameter uncertainty and process variability. Technical report, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA

Stettner M, Schrage DP (1992) An approach to tiltrotor wing aeroservoelastic optimization. In: 4th AIAA/USAF/NASA/OAI symposium on multidisciplinary analysis and optimization, Cleveland, Ohio. http://citeseer.ist.psu.edu/stettner92approach.html

Wrenn GA (1989) An indirect method for numerical optimization using the Kreisselmeier–Steinhauser function. Technical report CR-4220, NASA