Adaptive Constraint Aggregation for Structural Optimization Using Adjoint Sensitivities

Nicholas M. K. Poon
Graduate Student
mkpoon@utias.utoronto.ca

Joaquim R. R. A. Martins
Assistant Professor
martins@utias.utoronto.ca

University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto
Ontario M3H 5T6, Canada

Abstract

Constraint aggregation is the key for efficient structural optimization when using the adjoint method for sensitivity analysis. The most widely used constraint aggregation approach, the Kreisselmeier–Steinhauser function, can reduce the number of constraints and return a conservative estimate. However, this function reduces the accuracy of the optimization results. This inaccuracy is most prominent when constraints are active and increases with the number of active constraints. Using an adaptive approach, this undesirable characteristic is avoided by preserving the exact feasible region. The aggregation parameter is updated according to the constraint sensitivity, and a more accurate result is obtained without additional solver evaluations. This new approach significantly improves the accuracy of the optimization when a large number of constraints are active at the optimum. This improvement is illustrated by the weight optimization of a wing box structure.

Introduction

Design optimization is a growing research area that offers significant improvement on traditional design processes. For most engineering problems, design constraints that represent safety indices, performance boundaries and other limits on the end-product are necessary and must be handled during the optimization. However, handling large numbers of constraints increases the computational time and affects the convergence behaviour. In large scale problems, such as the multidisciplinary design optimization (MDO) of aircraft, approaches are needed to minimize these effects in order to obtain the optimum in reasonable time with adequate accuracy. In the context of large scale optimization, only gradient-based algorithms such as Sequential Quadratic Programming (SQP) that utilizes both function value and function gradient are viable to determine accurate results in oppose to the gradient-free algorithms. When using gradient-based algorithms, calculation of the function gradients can be the most time-consuming step in the optimization cycle. This fundamental issue can be resolved by efficient algorithms such as the analytic direct method, which is efficient for cases with few variables and many constraints or the analytic adjoint method is suited for computing the gradients of a few functions with respect to many variables [6, 12]. However, there is no efficient algorithm that is efficient in problems with large number of design variables and constraints simultaneously. To reduce discontinuities and the computational cost of the sensitivity analysis procedure, constraint aggregation is used to lump multiple constraints into a single composite function. With only a single or few composite constraints, the analytic adjoint method can compute the sensitivities efficiently.

The Kreisselmeier–Steinhauser (KS) function is a widely used constraint aggregation method for gradient-based optimization [6, 1, 11, 4] and has been implemented in the design of a supersonic business jet (SSBJ) [6] and a hypersonic civil transport (HSCT) [2]. A representation of part of the MDO framework used for the SSBJ design is shown in Figure 1 with the KS module highlighted. While using KS function provides conservative result in design cases that had been studied, this also results in inaccurate results in design cases that have constraints that are active at the optimum which is very often the case in structural optimization.

The objective of this study is to derive an adaptive KS function for structural optimization that uses adjoint sensitivity calculations. We will show that with increasing number of active constraints, the error of the optimization results increases and eventually results in unacceptable inaccuracies. A sample optimization problem is used to illustrate the trend in error increment. A more elaborate sample considers the optimization of a wing structure is used to demonstrate the application of the adaptive approach and the improvement in accuracy. This optimization uses SNOPT [6] as the optimizer and FEAP [3] as the computational structural mechanics solver.
Structural Optimization using Adjoint Sensitivities

Within the existing MDO framework, the main utility of the KS function is to make the adjoint sensitivity calculation of stress constraints more efficient. While our goal is to closely examine the optimization with adjoint sensitivity of the composite constraint, only the structural optimization shown in Figure 2 will be studied. Consider a typical structural optimization problem,

\[
\begin{align*}
\text{Minimize} & \quad W(x) \\
\text{w.r.t.} & \quad x_n \in \mathbb{R}^n, \\
\text{Subject to} & \quad g_j \leq \sigma_{\text{Yield}}, \quad j = 1, \ldots, N_{\text{elements}}
\end{align*}
\]

where \( W \) is the structural weight, \( x \) is the vector of design variables and \( g_j \) are the stress constraints. In most cases, where the finite element method is used, the constraints are imposed on each element. In high-fidelity optimization, there may be \( O(10^6) \) elements and the number of constraints can 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.

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 in the structural optimization problem is the displacement vector, \( u \), which is obtained by solving the governing equations,

\[
R = Ku - F = 0
\]

The adjoint method is based on the application of chain rule to the differentiation of the function of interest, \( I(x,u) \), which is a function of the design variables, \( x \) and the state variables, \( u(x) \). The functions of interest in the structural optimization problem are the stress constraints which have dimensions \( 1 \times N_{\text{elements}} \). The adjoint equations are,

\[
\begin{align*}
\frac{dI}{dx} &= \frac{\partial I}{\partial x} + \frac{\partial I}{\partial u} \frac{du}{dx} \\
\frac{dR_k}{dx} &= \frac{\partial R_k}{\partial x} + \frac{\partial R_k}{\partial u} \frac{du}{dx} = 0 \\
\frac{du}{dx} &= -\frac{\partial R_k}{\partial u} \frac{1}{\partial R_k} = \frac{\partial R_k}{\partial x}
\end{align*}
\]

Therefore,

\[
\frac{dI}{dx} = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial u} \left( \left[ \frac{\partial R_k}{\partial u} \right]^{-1} \right) \frac{\partial R_k}{\partial x}
\]
From Equation (4), we can see that the most costly step is the calculation of the adjoint vector, \( \psi \), and factorization is used to solve the adjoint equations,

\[
\frac{\partial R_k}{\partial u} \psi = -\frac{\partial I}{\partial u} \tag{5}
\]

For each function of interest, a different adjoint vector is required. As a result, the adjoint method is dependent to the number of functions of interest, and it is the most efficient when the system has one or few outputs. Thus the application of the adjoint method to the problem in Equation (1) cannot utilize its advantages.

By assuming that the constraints, \( g_j \), are continuous — but not necessarily being continuously differentiable — the constraints can be aggregated into a composite constraint. Through this reformulation, multiple constraints are aggregated into the Kreisselmeier–Steinhauser (KS) function and the modified optimization problem becomes the following.

\[
\text{Minimize } W(x)
\]

w.r.t. \( x_n \in \mathbb{R}^n \),

Subject to \( KS[\sigma_j(x)] \leq 0 \),

\( j = 1, \ldots, N_{\text{elements}} \)

In this modified optimization problem, the KS function represents an n-dimensional constraint function defining a conservative estimate of the maximum of all the constraints. Figure 3 illustrates the role of the KS function in the optimization cycle.

**Figure 3:** Structural optimization with constraint aggregation

**Constraint Aggregation Using Kreisselmeier–Steinhauser (KS) Function**

The KS function was first presented by G. Kreisselmeier and R. Steinhauser [5]. The purpose was to minimize the gains in controllers of a closed loop feedback control system, while having the output specifications as performance indices (constraints). This was achieved by performing an iterative search. The performance indices were combined into a single one at each iteration by the logarithm of an exponential series, the KS function.

This procedure was later categorized as a sequential unconstrained minimization technique (SUMT) where the pseudo-objective is the KS function that combines the objective functions and inequality constraints into one composite function. The function also involves a "draw-down" factor or aggregation parameter, \( \rho \), which is analogous to the penalty factor in penalty methods used to perform constrained optimization. The function can be written as,

\[
KS(f(x), g_j (x)) = \frac{1}{\rho} \log_e \left( \sum_{i}^{n_f} e^{\rho f(x)} + \sum_{i}^{n_g} e^{\rho g_j(x)} \right) \tag{7}
\]
This defines an envelope surface in the n-dimensional design space that replaces the objective function and the constraint. This envelope surface is $C_1$ continuous and represents a conservative estimate of the maximum among the set of functions [15]. This formulation allows optimization with multiple objectives and constraints, and was widely used before direct constrained optimization techniques became popular.

Single discipline optimization applications include aerodynamic shape optimization with unstructured grids for turbulent flows [4] and tiltrotor wing aeroservoelastic optimization for control system design [13]. As for multidisciplinary optimization, there are examples ranging from the MDO of cooled gas turbine blades [14] to the automotive design optimization using structural dynamic and fatigue sensitivity derivatives [16]. The KS function can also be used as a constraint aggregation technique that represents large number of constraints as a single composite function, which can be written as,

$$KS(c_j(x)) = \frac{1}{\rho} \log_e \left( \sum_{i}^{n_c} e^{\rho c_j(x)} \right)$$

(8)

This definition is mostly used with conjunction with the sequential quadratic programming (SQP) and trust region methods. Some of the documented applications include structural optimization with multiple load-cases [2], high-fidelity aero-structural optimization [6] and design optimization of a high-speed civil transport aircraft [1].

The properties of the KS function as derived in [10] are listed below.

- **Property 1:** $KS(x, \rho) \geq \max(g_j(x))$ for $\rho > 0$
- **Property 2:** $\lim_{\rho \to \infty} KS(x, \rho) = \max(g_j(x))$ for $j = 1, \ldots, N_g$
- **Property 3:** $KS(x, \rho_2) \geq KS(x, \rho_2)$ for $\rho_1 > \rho_2 > 0$
- **Property 4:** $KS(x, \rho)$ is convex, if and only if all constraints are convex.

These properties are vital for the KS function to be considered as a desirable constraint aggregation technique for nonlinear programming. Property 1 through 3 indicate that KS function is a conservative overestimation for constraints defined as $g_i \leq 0$ where a positive value will be returned if a constraint is violated or close to be violated. Also, the conservative nature is determined by the magnitude of the aggregation parameter, $\rho$. As $\rho$ increases, KS function approaches the maximum constraint at the current design point, $x$. Property 4 implies the use of KS function does not alter the convexity of a problem. Therefore, if the original problem is convex where the objective and constraint functions are all convex, then the modified problem with constraints aggregated into KS function remains convex.

**Formulation of the KS Function**

The formulation of the KS function was originally derived for inequality constraints [5].

$$KS(g_j(x)) = g_{max} + \frac{1}{\rho} \log_e \left[ \sum_{j}^{N_g} e^{\rho (g_j - g_{max})} \right]$$

(9)

The magnitude of the KS function value determined at a particular design point is within the limit bounded by the upper and lower values in the expression,

$$g_{max} < KS < g_{max} + \frac{\log_e (N_h + N_g)}{\rho}$$

(10)

The lower bound (most accurate) value is $g_{max}$, the maximum value among all individual constraints at a particular design point. The upper bound (least accurate) value is defined by the right hand side expression. The “draw-down” factor, $\rho$, that determines the difference between the KS function and the maximum value of the constraint, $g_{max}$, at any particular design point, $x$. As $\rho$ approaches infinity, the KS function becomes equivalent to $g_{max}$, the maximum of the all the constraints at any given $x$.

From Equation (10), we can calculate the maximum error for a particular $\rho$ value, or the $\rho$ value from a chosen maximum error. Although machine zero error is desirable and can be achieved theoretically by having $\rho$ be large enough, obtaining second order derivatives numerically at the design point where the constraints intersect poses difficulties if step-size is not small enough. This is illustrated in Figure 4. At $x \approx 6.7$, estimating the Hessian of the KS function yields an ill-conditioned matrix when $\rho$ is too large, which causes numerical difficulties. Therefore, $\rho = 50$ is usually a reasonable value that has a maximum error of $\approx 0.03$ for 2 constraints, and is often used [7].
In the following single-variable examples, the application of the KS function as constraints aggregation and the effect of increasing $\rho$ for inequality constraints can be visualized. Consider the following convex inequality constraints,

$$g_1 (x) = 5 \log_e (x) - \frac{x^5}{5} - 4 \leq 0$$
$$g_2 (x) = \frac{x^2}{40} + \frac{x}{5} - 2 \leq 0$$

The KS function of these two constraints will form an envelope as shown in Figure 5.

Note that the convexity of the KS function is inherited from the original constraints and therefore the KS function will be convex if and only if the constraints are all convex in the domain considered. Also, as $\rho$ increases, KS became closer resemble to the maximum constraint, $g_{max}$. This is especially apparent at the intersection of the two constraints.
Adaptive Approach for Constraint Aggregation

The original formulation can be classified as non-adaptive as it carries the predetermined aggregation parameter $\rho$ throughout the optimization. Although this parameter (typically 50 or less) is selected for good numerical handling, the function will return a conservative value at any point in the domain. However, this widely adopted approach results in inaccuracies to the optimization results as shown in Figure 6 for an analytic optimization problem [9]. This problem is selected because it is representative of the behaviour at the vicinity of local optima that is observed in structural optimization with stress constraints.

![Figure 6: Relative error in optimization result using traditional KS function with $\rho = 50$](image)

The error increases to 10% for only 100 constraints and 15% for 2500 constraints. The trend of this error becomes prohibitive in practical finite element models, which attributes with more than 10000 elements and thus 10000 stress constraints. Also note that the relationship deduced from Figure 6 agrees with the maximum error determined from Equation (10) and is an important bound to be considered.

This sub-optimal result suggests that the traditional approach poses difficulties to the optimizer that prevent it for converging to the true optimum. Possible reason for this error is due to the feasible region defined by the KS function does not contain the true optimum. As shown in Figure 7 for a single variable example, where $F(x)$ is the objective, $g_1(x)$ and $g_2(x)$ are the two constraints, the objective function changes more rapidly across the domain than the constraints. While the traditional KS function is subjected to the curvature of the original constraints, KS defines a smaller feasible region than the region due to the original constraints. Thus, the actual feasible region is not visible to the optimizer if a traditional KS function is used. This becomes more significant when the optimum is located at the intersection of constraints.

![Figure 7: Feasible region reduction with traditional KS](image)

According to Property 2, the KS function tends to the exact value as the aggregation parameter approaches infinity. However, an aggregation parameter of large magnitude may lead to numerical overflow in the course of the optimization. Therefore, aggregation parameter should only be increased as required. This can be achieved by defining a nominal aggregation parameter which is typically 50 or less.
parameter at the beginning of the optimization, e.g. $\rho = 50$, and increase it as needed according to the sensitivity of KS with respect to the aggregation parameter, $(KS' = dKS/d\rho)$ at the current design points. 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 non-zero and reaches a maximum when the design point is at the constraints intersection, dictated by Equation (10). Figures 8 and 9 show the KS sensitivity with increasing $\rho$ for design points at and nearby constraint intersection respectively.

\begin{align*}
\frac{\log KS'_i - \log KS'_d}{\log \rho_1 - \log \rho_c} &= \frac{\log KS'_i - \log KS'_c}{\log \rho_d - \log \rho_c} \\
(11)\\
\log \rho_d &= \log \left( \frac{KS'_d}{KS'_c} \right) \left[ \log \left( \frac{KS'_i}{KS'_c} \right) \right]^{-1} \log \left( \frac{\rho_1}{\rho_c} \right) + \log \rho_c \\
(12)
\end{align*}

This relationship is derived from Figure 8. The subscript $i$ represents a value calculated at a finite step from the current point. In our application, $\rho_1 = \rho_c + 1 \times 10^{-3}$. By rearranging Equation (11), the desired value, $\rho_d$, is expressed as,

In our implementation, $\rho_c$ is set to 50, which is a conservative estimate recommended in [1, 10, 15]. During the optimization process, $\rho_c$ is kept constant and $\rho_d$ is used for the calculation of $KS$ and $dKS/d\rho$ if $KS'_i$ does not meet the prescribed tolerance. Also, $KS'$ is calculated using complex-step method [7] in order to obtain accurate results. The resulting algorithm is illustrated in Figure 10.

**Algorithm AKS (Adaptive Kreisselmeier–Steinhauser(KS) Function)**

Given:
- Constraint function: $g_j(x)$ for $j \in (1, ..., N_g)$
- Initial aggregation parameter: $\rho_c$
- Desired KS sensitivity: $KS'_d$

The following sequence is executed when a constraint evaluation is requested by the optimizer.
1. Initialize the function with the given parameters
2. Calculate $KS'$ at current design point $x$ and $\rho_c$, with the constraint functions $g_j$
3. If $KS'_c < KS'_d$, return KS value; otherwise, continue
4. Calculate $\rho_d$ using Equation (12)
5. Calculate KS using $\rho_d$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8.png}
\caption{Plot of $KS'$ vs. $\rho$ at intersection}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig9.png}
\caption{Plot of $KS'$ vs. $\rho$ near intersection}
\end{figure}
Results

In order to demonstrate the application of the adaptive approach, a weight optimization study is conducted for a wing structure modelled with multiple spars and ribs with a total of 45 tube elements as illustrated in Figure 11. The finite element model uses a modified 3D frame elements that account for 1 translational degree of freedom (vertical) and 2 rotational degrees of freedom (axial and transversal) at each node, thus each element has a total of 6 degrees of freedom.

![Figure 11: FEM model of the wing structure](image)

The diameters, weight and stress are the design variables, objective function and constraints respectively, and the optimization problem can be described as following.

\[
\text{Minimize} \quad W(x) \\
\text{w.r.t.} \quad x_n \in \mathbb{R}^n, \quad n = 1, \ldots, 45 \\
\text{Subject to} \quad g_j = \frac{\sigma_j (x, u)}{\sigma_{Yield}} - 1 \leq 0 \quad j = 1, \ldots, 45
\]

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

Optimization Settings

The initial set of design variables is taken as unity in order to maintain the generality of the demonstration. As for constraint handling, different methods are considered. These methods are the original KS function (KSC), the maximum function (MAX) and the new adaptive KS function (AKS), which will be compared with the reference case that considered all constraints separately (IC). The KSC and AKS formulation are introduced in previous sections. The MAX formulation is similar to the exact penalty term as in SUMT [8] which selects the most violated constraint at the current design point. MAX here is considered as one of the constraint aggregation methods and forms a n-dimensional composite function similar to KSC. However, its function is not continuously differentiable at constraints intersections; therefore its numerical gradient may be ill-conditioned and not suitable for use in numerical optimization.
Computational Cost

The adaptive constraint aggregation method was implemented in a Python based programming interface that connects the optimizer, SNOPT, and the CSM solver, FEAP. SNOPT is an active-set sequential quadratic programming (SQP) based optimizer. The computational cost for function and gradient evaluations of the various constraint handling methods are summarized in Figure 12.

<table>
<thead>
<tr>
<th>Constraint Handling Method</th>
<th>Individual Function</th>
<th>Maximum Constraint</th>
<th>KS Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Difference</td>
<td>46</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>Central Difference</td>
<td>91</td>
<td>91</td>
<td>91</td>
</tr>
<tr>
<td>Complex Step</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Derivative</th>
<th>Individual Function</th>
<th>Maximum Constraint</th>
<th>KS Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Difference</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Central Difference</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Complex Step</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 12: Computational cost of the constraint function and sensitivity

In analogy to previous sections, the computational cost of adjoint sensitivity method comes from the solution of the adjoint vector for each constraint. Therefore when constraints are aggregated, computational cost is reduced for the corresponding adjoint equation size.

While the adjoint sensitivity using partial derivatives by both forward difference and the complex step have good performance, the accuracy of the former has a minimum error of $10^{-6}$ while the later achieves $10^{-14}$ as shown in Figure 13, which is a lot more reliable. Therefore adjoint sensitivity using partial derivatives computed with the complex step method is used in the optimization study.
Optimization Results

Figure 14 shows the initial structure with design variables set to unity and the optimization results using different constraint handling methods. The result by using maximum constraint is infeasible with some of the elements exceed the prescribed stress. Optimization using the KS constraint and the adaptive approach is feasible and the former is a conservative design that element stress are about 80% of the prescribed yield stress while the adaptive approach leads to a more optimal design comparing to the reference case.

![Initial Structure](image)

- **Initial Structure**
  - Weight: 2741.351 kg

![Optimized Structure](image)

- **Optimized Structure (IC)**
  - Weight: 183.500 kg

- **Optimized Structure (MAX)**
  - Weight: 247.152 kg

- **Optimized Structure (KS: ρ=50)**
  - Weight: 194.224 kg

- **Optimized Structure (AKS)**
  - Weight: 183.076 kg

Figure 14: Optimization Results

Convergence History

The convergence history of the optimization is shown in Figure 15 for different types of constraint handling with the reference case being the solution of the optimization by considering the constraints individually.

![Convergence History](image)

From Figure 15, the maximum constraint approach demonstrated poor convergence behaviour, which requires extended operational time and low accuracy. The final result is in fact inadmissible with violating constraints. This is due to its lack of first order continuity and cause the optimizer can not make proper estimate of the Lagrange multiplier, which lead to poor performance when using the active-set SQP optimizer.
The original KS function however, converges to optimum in about one third the reference time and the result is feasible to a relative difference of 6%. Since the optimum is located at the intersection of active constraints, using constant aggregation parameter leads to reduction of the feasible region as seen by the optimizer and results in an inaccurate optimum.

On the other hand, when compared using the adaptive approach, the optimization converges in 75% the reference time and the optimum is accurate to ±0.1% to the reference case.

**Optimization Performance vs. Problem Size (Number of Constraints)**

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

In Figure 16, the white coloured symbols represent feasible optimum and the black symbols represent infeasible optimization results. Optimization with the adaptive approach always returns feasible optima differ from the reference by only ≈ 0.1%.

**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 more accurate result without additional solver evaluations and is more robust than using large magnitude of the aggregation parameter in the original formulation. From the sample weight optimization results, the adaptive approach demonstrated a better performance than the reference case where constraints are considered separately. Its accuracy is significantly improved when compared to the original KS formulation and matches the reference case.
References


