

## Efficient multi-partition topology optimization

Koppen, S.; Langelaar, M.; van Keulen, F.

**DOI**

[10.1016/j.cma.2022.114829](https://doi.org/10.1016/j.cma.2022.114829)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

Computer Methods in Applied Mechanics and Engineering

**Citation (APA)**

Koppen, S., Langelaar, M., & van Keulen, F. (2022). Efficient multi-partition topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 393, Article 114829. <https://doi.org/10.1016/j.cma.2022.114829>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Efficient multi-partition topology optimization

S. Koppen\*, M. Langelaar, F. van Keulen

*Department of Precision and Microsystems Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*

Received 3 November 2021; received in revised form 21 January 2022; accepted 28 February 2022

Available online xxxx

## Abstract

In topology optimization, the state of structures is typically obtained by numerically evaluating a discretized PDE-based model. The degrees of freedom of such a model can be partitioned in free and prescribed sets to define the boundary conditions. A multi-partition problem involves multiple partitions of the same discretization, typically corresponding to different loading scenarios. As a result, solving multi-partition problems involves multiple factorization/preconditionings of the system matrix, requiring a high computational effort. In this paper, a novel method is proposed to efficiently calculate the responses and accompanying design sensitivities in such multi-partition problems using static condensation for use in gradient-based topology optimization. A main problem class that benefits from the proposed method is the topology optimization of small-displacement multi-input–multi-output compliant mechanisms. However, the method is applicable to any linear problem. We present its formulation and an algorithmic complexity analysis to estimate computational advantages for both direct and iterative solution methods to solve the system of equations, verified by numerical experiments. It is demonstrated that substantial gains are achievable for large-scale multi-partition problems. This is especially true for problems with both a small set of number of degrees of freedom that fully describes the performance of the structure and with large similarities between the different partitions. A major contribution to the gain is the lack of large adjoint analyses required to obtain the sensitivities of the performance measure.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

**Keywords:** Static condensation; Topology optimization; Computational efficiency; Sensitivity analysis; Partitioning; Adjoint

## 1. Introduction

Finding a solution to large scale gradient-based topology optimization problems in a nested analysis and design approach requires a significant amount of computational effort due to the need to perform one or multiple expensive simulations per design iteration [1]. These simulations are often based on solving discretized governing partial differential equations with many Degrees of Freedom (DOFs). A solution to such a linear system of equations is an assignment of values to the unknown variables at each DOF for given Boundary Conditions (BCs), such that all the equations are satisfied simultaneously. In this setting, we call this set of variables the state, which may represent nodal values such as, e.g., temperature, displacement, pressure or velocity.

One may opt for a direct or iterative solution method to solve the involved system of linear equations; both methods consist of two subsequent steps. Direct methods require the (generally expensive) construction of

\* Corresponding author.

E-mail address: [s.koppen@tudelft.nl](mailto:s.koppen@tudelft.nl) (S. Koppen).

factorization, and subsequently find the exact solution via comparatively inexpensive back-substitutions [2]. In contrast, iterative methods require the construction of a preconditioner, and subsequently generate a sequence of improving approximate solutions until convergence [3,4]. Relative cost of preconditioner construction and the iterative solution process depends on many factors, such as the type of preconditioner and condition number. For conciseness, we will hereafter denote the construction of factorization/preconditioner simply by *preprocessing* and the back-substitution/iterative method simply by the *solve*.

A priori to finding the solution to the governing equations, the DOFs are generally partitioned in two unique sets based on the applied BCs. One may either specify (i) the magnitude of the solution at a DOF, that is the state is prescribed, or (ii) the magnitude of the applied discretized load, and thus the state is free. These types are complementary—all DOFs for which the state is free will have an applied load (although possibly with zero magnitude), and vice versa.

As Rozvany et al. [5] pointed out almost three decades ago, real-world optimization problems generally involve multiple scenarios. We define a scenario as an engineering description of a specific loading condition, see for example the four scenarios depicted in Fig. 1. Examples of optimization problems that typically involve multiple scenarios are problems with (i) uncertainties in the intended use of the product, (ii) multiple use cases, such as multi-input–multi-output compliant mechanisms, (iii) a strict set of requirements (as obtained for example from the user), see *e.g.* Cavazzuti et al. [6], and (iv) problems that consider—in addition to the critical use case scenarios—variations in boundary and loading conditions originating from manufacturing, assembly, transportation and testing of the product. Each scenario is translated to BCs, *i.e.* prescribed state(s) and applied load(s), that represent the scenario as good as possible. In principle, each scenario comprises a unique partitioning of DOFs. However, scenarios of which only the magnitude of the applied BCs vary, but the type remains the same, share equal partitioning.

In this work, we define an analysis as the process of finding a solution to a properly constrained set of discretized governing equations—that is finding the state  $\mathbf{u} \in \mathbb{R}^n$ , with  $n \in \mathbb{N}$  the number of nodal DOFs. Each analysis relates to a single scenario, and thereto also comprises a unique partitioning of DOFs. A combination of analyses with equal partitioning is hereafter called an analysis set. A single analysis set thus requires finding the states to multiple loads (possibly with varying magnitude) with the same partitioning of DOFs. These different states can generally be efficiently found, since only a single preprocessing process is required [7].<sup>1</sup>

Topology optimization problem formulations involving responses (objective and constraints) that depend on the states of multiple analyses sets are called Multi-Partition Optimization Problems (MPOPs). In contrast to problems involving a single partition, in MPOPs the preprocessing cannot be reused. This is evident in the design of, for example, multi-input–multi-output compliant mechanisms (see *e.g.* Liu and Korvink [10] and Jin et al. [11]).

For clarification, consider a response, hereafter denoted by  $g \in \mathbb{R}$ , that depends on the state(s) of  $a \in \mathbb{N}$  analysis sets. Thus,  $g = g[\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(a)}]$ , with  $\mathbf{U}^{(i)} \in \mathbb{R}^{n \times l^{(i)}}$  and  $l^{(i)} \in \mathbb{N}$  a positive natural number of analyses contained in analysis set  $i$ , thus  $\mathbf{U}^{(i)} := [\mathbf{u}_1^{(i)}, \dots, \mathbf{u}_{l^{(i)}}^{(i)}]$  with each state  $\mathbf{u}^{(i)} \in \mathbb{R}^n$ . The states of an analysis set  $\mathbf{U}^{(i)}$  implicitly depend on the design variables  $\mathbf{x} \in \mathbb{R}^N$  via the discretized governing equations, that is

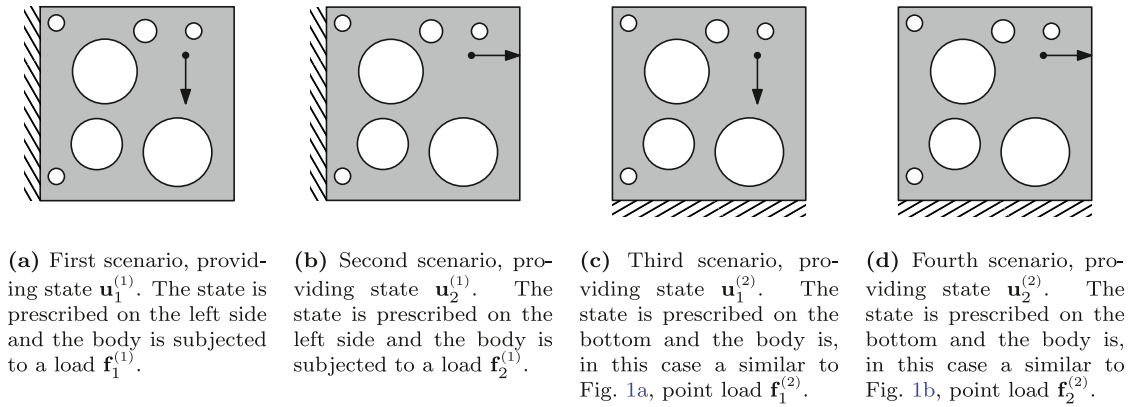
$$\mathbf{K}[\mathbf{x}] \mathbf{U}^{(i)} = \mathbf{F}^{(i)}, \quad \forall i \in \mathbb{A} \quad (1)$$

with the index set of all analyses sets  $\mathbb{A} := \{i \in \mathbb{N} \mid i \leq a\}$ , the design dependent system matrix  $\mathbf{K}[\mathbf{x}] \in \mathbb{R}^{n \times n}$  and loads  $\mathbf{F}^{(i)} \in \mathbb{R}^{n \times l^{(i)}}$ . Hereafter we assume the system matrix is symmetric and, without loss of generality, assume the applied loads are design independent. Note that, in Eq. (1), the states and loads of analysis set  $i$  contain all DOFs; both free and prescribed states as well as applied and reaction loads.

A simple example of an MPOP is the optimization of a structure considering four scenarios, schematically shown in Fig. 1. Note that between the four scenarios the set of DOFs for which the state is prescribed differs. Therefore, in this example, two unique analysis sets exist, comprising of two analyses each. The response function  $g$  of this artificial problem depends on the states of these two unique analysis sets, *i.e.*  $g[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}]$ , where for this problem  $\mathbf{U}^{(i)} = [\mathbf{u}_1^{(i)}, \mathbf{u}_2^{(i)}]$  are the states (in this case the displacement) of analysis set  $i$ .

In spite of the fact that MPOPs are relevant for many applications, the computational efficiency of both the forward and sensitivity analyses involved in such problems has, to the best knowledge of the authors, not been investigated yet. With an increasing number of analysis sets, the computational effort to solve MPOPs increases quickly. Since the preprocessing used in the analysis occurring in MPOPs is partition-specific, in a straightforward

<sup>1</sup> For very large number of loads one can opt for efficient sampling schemes, see *e.g.* Zhang et al. [8,9].



**Fig. 1.** A structural topology to be optimized for four different scenarios, as shown in Figs. 1(a) to 1(d). As a result, this Multi-Partition Optimization Problem (MPOP) consists of two unique analysis sets, both comprising two analyses.

approach the computational effort of solving these problems scales with the number of analysis sets considered. On the contrary, within reasonable limits, the number of analyses per set is of less importance, given the fact that a solve requires negligible computational effort as compared to the preprocessing.

The computational efficiency of an analysis in the context of structural optimization can be increased by efficient iterative solution strategies (see *e.g.* Evgrafov et al. [12], Amir et al. [4], Aage et al. [13] and Wu et al. [14]), or approximated reanalysis, see *e.g.* Kirsch [15], Amir et al. [1] and Amir [16]. Alternatively, one can opt for model-order reduction techniques, which reduce the computational effort of both analysis and optimization, see *e.g.* Yoon [17] and Choi et al. [18]. Typically, however, model-order reduction techniques introduce a loss of accuracy. One special reduction method is what is called static condensation amongst engineers, for which the reduction is *exact* [19,20]. As will be shown hereafter, in particular in MPOPs the use of static condensation is attractive. Moreover, as static condensation, similar to an analysis as previously defined, relies on finding the solution to a set of linear system of equations, it can also be combined with efficient iterative solution techniques and/or approximated reanalysis.

Static condensation was originally developed by Guyan [19] and Irons [20], hence the common nomenclature Guyan–Irons reduction scheme. The term ‘static’ was added to differentiate from its application in dynamic analysis, for which the reduction scheme is no longer exact. Static condensation is regularly applied in the process of substructuring, also known as the Schur complement domain decomposition method [21–23]. This technique is based on the decomposition of the original structure into one or multiple sub-structures, often referred to as super-elements. The dimensionality reduction is achieved by a procedure that is closely related to Kron’s method of tearing and Gaussian elimination [24,25].

Consider the continuum from Fig. 2(a). After discretization of the governing equations in  $n$  DOFs, as shown in Fig. 2(b), the behaviour of the system can be described by the set of linear equations given by

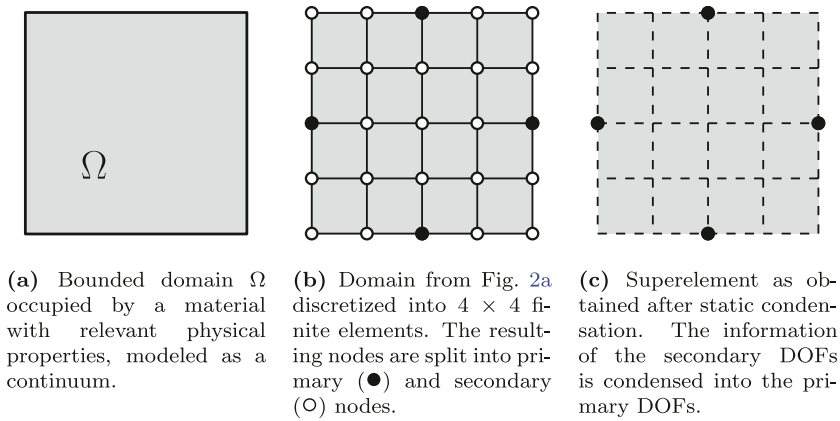
$$\mathbf{K}[\mathbf{x}]\mathbf{u} = \mathbf{f}, \quad (2)$$

where, for simplicity, we consider a single analysis set consisting of a single analysis and thus  $\mathbf{u} \in \mathbb{R}^n$  and  $\mathbf{f} \in \mathbb{R}^n$ .

The DOF index set  $\mathbb{D} := \{i \in \mathbb{N} \mid i \leq n\}$  can be split in subset  $\mathbb{M} \subset \mathbb{D}$  containing all indices of the primary DOFs, and its complement  $\mathbb{S} := \mathbb{D} \setminus \mathbb{M}$  containing all indices of the secondary DOFs. The primary DOFs are those DOFs contained in the condensed model, whereas the secondary DOFs are eliminated. After partitioning accordingly, one may rewrite the governing equations from Eq. (2) using oversets ( $\wedge$ ) and ( $\vee$ ) to represent the properties related to primary and secondary DOFs, respectively, as

$$\begin{bmatrix} \overset{\wedge}{\mathbf{K}} & \overset{\vee}{\mathbf{K}} \\ \overset{\vee}{\mathbf{K}} & \overset{\wedge}{\mathbf{K}} \end{bmatrix} \begin{bmatrix} \overset{\wedge}{\mathbf{u}} \\ \overset{\vee}{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \overset{\wedge}{\mathbf{f}} \\ \overset{\vee}{\mathbf{f}} \end{bmatrix}, \quad (3)$$

where the principal sub-matrix  $\overset{\wedge}{\mathbf{K}}$  is *nonsingular*, such that  $\overset{\wedge}{\mathbf{K}}$  is invertible [26]. This partitioning is completely independent of the partitioning in free (f) and prescribed (p) DOFs. In case we are solely interested in the state



**Fig. 2.** Step-wise illustration of static condensation. For simplicity, and without loss of generality, the domain is chosen to be a 2D square and discretized using a structured mesh of  $4 \times 4$  quadrilateral finite elements. For simplicity of visualization, we consider all DOFs of a single node to be either primary or secondary DOFs. Primary nodes are indicated by a disc (●), whereas secondary nodes are indicated by a circle (○).

of the primary DOFs, we can write the secondary state  $\check{\mathbf{u}}$  in terms of the primary state  $\hat{\mathbf{u}}$  using the second row of Eq. (3), which yields

$$\check{\mathbf{u}} = -\check{\mathbf{K}}^{-1} \check{\mathbf{K}} \hat{\mathbf{u}}, \quad (4)$$

assuming  $\check{\mathbf{f}} = \mathbf{0}$  for simplicity (the general case will be treated in Section 2). Subsequently,  $\check{\mathbf{u}}$  can be eliminated by substitution of Eq. (4) in the first row of Eq. (3), which gives

$$\left( \hat{\mathbf{K}} - \check{\mathbf{K}} \check{\mathbf{K}}^{-1} \check{\mathbf{K}} \right) \hat{\mathbf{u}} = \hat{\mathbf{f}}. \quad (5)$$

The result of this condensation is a reduced system matrix  $\tilde{\mathbf{K}} \in \mathbb{R}^{m \times m}$ , with the number of primary DOFs  $m := |\mathbb{M}|$ , that exactly describes the behaviour of the full-order model. This reduced system matrix follows as

$$\tilde{\mathbf{K}} := \hat{\mathbf{K}} - \check{\mathbf{K}} \check{\mathbf{K}}^{-1} \check{\mathbf{K}}, \quad (6)$$

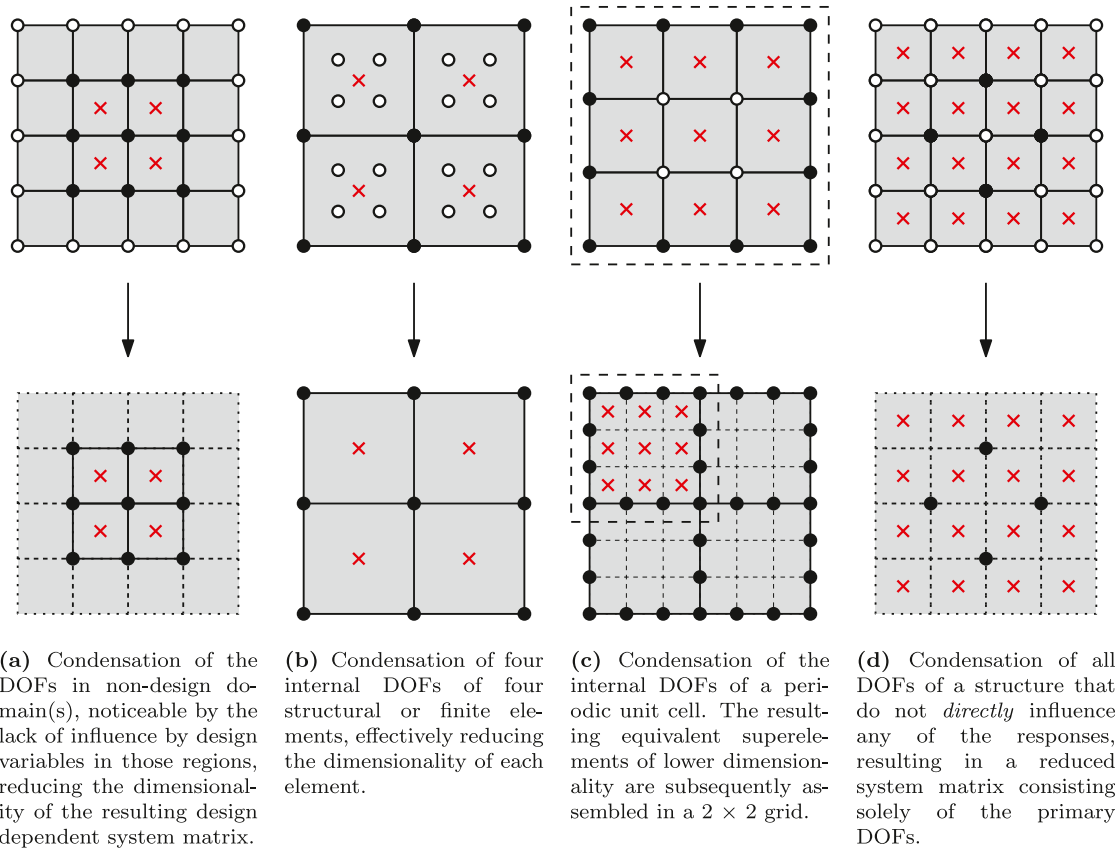
and was originally proposed as the “Schur complement of  $\check{\mathbf{K}}$  in  $\mathbf{K}$ ”, first introduced in this form and notation by Haynsworth [27]. The eponymous adjective “Schur” was chosen by Haynsworth because of the first use of the complement in the “Schur determinant lemma” published by Schur [28].

Static condensation has been applied in the field of gradient-based structural optimization in a variety of settings. Contributions include formulations that condense the

- DOFs within non-design domain(s) [29–31],
- internal DOFs of (complex) finite elements or components [32–35],
- internal DOFs of repetitive unit cell(s) [36,37], and/or
- DOFs of a structure that do not occur in the responses, *e.g.* all DOFs except input and output in a compliant mechanism [38,39].

Fig. 3 schematically illustrates the use of static condensation in each of these applications, which are discussed in more detail hereafter.

Early studies by Botkin and Yang [29], Gangadharan et al. [30] and Yang and Lu [31] aim to reduce the computational effort of both the Finite Element Analysis (FEA) and/or the sensitivity analysis of 3D structural optimization problems by static condensation of specific sub-domains. The topology in those domains does not depend on the design variables and hence remain unchanged during optimization, as shown in Fig. 3(a). Since only a small part of the structure is subject to change, significant computational efficiency increase can be achieved if the dimensionality of the problem is reduced before the optimization, such that repetitive analysis of the non-design domain(s) is avoided. Since these early studies, this use of condensation in optimization has become commonplace and is also available in commercial software packages.



**Fig. 3.** Four different settings in which static condensation can be used in structural optimization. Top and bottom figures illustrate discretized domains before and after the condensation process, respectively. Similar to Fig. 2, discs (●) and circles (○) denote primary and secondary nodes, respectively. Red crosses (×) indicate the location of design variables, which only locally influence the design.

In the fields of geometrically non-linear, high-resolution and multi-component layout structural optimization, static condensation is effectively applied to eliminate internal DOFs from finite elements before global assembly, as schematically shown in Fig. 3(b). This process reduces element complexity and decreases the dimensionality of the assembly system matrix, see *e.g.* Yoon et al. [32], Xia et al. [33], Groen et al. [34] and Amir and Amir [35].

In addition to the aforementioned, the computational effort can be further reduced in case the structure is periodic. Since all repeated cells share the same topology, only static condensation of a single cell is required, as shown in Fig. 3(c). The system matrix of the cell is condensed to reduce the dimensionality of the assembled system matrix that forms the periodic structure, see *e.g.* Zhang and Sun [36] and [37].

A rather underexposed application of the static condensation procedure is its use in a unique set of optimization problem formulations of which the responses target a small set of DOFs, see Fig. 3(d). The state itself will generally depend on the full set of design variables. Representative examples are studies proposing problem formulations for the synthesis of compliant mechanisms with desired kinematics [38,40,41]. The added value of using static condensation in these formulations is the natural way of describing the performance of the structure as a function of a limited set of primary DOFs. In these formulations, the scenarios generally consist of a large number of DOFs that share the same type of BCs.

Although the usefulness of static condensation in FEA has been common knowledge, the usefulness in sensitivity analysis and the impact on the computational efficiency of structural optimization problem formulations has been unexplored so far. Gangadharan et al. [30] acknowledge and demonstrate the potential usefulness of static condensation to reduce the computational effort in the sensitivity analysis for problems in which the design variables only affect a small part of the domain. However, the usage of static condensation shows high potential in terms

of computational effort for solving MPOPs as well, as it could reduce the cost of both the forward and sensitivity analysis, as will be shown in Section 2. The findings in the present work explore the potential in this field, ready to be applied to real-world problems.

This paper presents a novel method to efficiently solve MPOPs of which the responses involve a relatively small set of DOFs, see Fig. 3(d). The method exploits the concept of static condensation in a way not reported before. We analyse and discuss the method's characteristics and limitations, provide a comparison between the proposed method and the state-of-the-art, and demonstrate the computational efficiency by numerical examples.

This work is focused on gradient-based topology optimization problems, involving linear state equations, solved via nested analysis and design. In addition, we assume (quasi-)static design independent loading conditions. The method operates on any square and symmetric system matrix, *e.g.* as obtained by assembly of 2D or 3D finite elements.

Note that it is not per se true that analyses of the reduced-order model are cheaper than analyses of the full-order model under all conditions. The reduced system matrix is not only of reduced dimensionality, but also non-sparse. This loss of sparsity may impair any efficiency increase due to dimension reduction. Hence, it is critical to understand under what conditions to apply static condensation in the context of gradient-based structural optimization.

A generalized mathematical description of the method and corresponding sensitivity analyses will be given in Sections 2 and 3. Both direct and adjoint sensitivity analysis aspects will be discussed. The description forms the basis for the formulation of a computational effort estimate. This estimate is based on the algorithmic complexity as a function of both the number of required analyses and the reduction of number of DOFs. The resulting estimation of computational efficiency with respect to a straightforward approach, *i.e.* performing all analyses without condensation, is outlined in Section 4. The computational effort estimate is verified by numerical experiments. Section 5 describes a comparison of computational efficiency for a variety of relevant examples. For simplicity, the proposed method is demonstrated by, but by no means limited to, 2D single-physics static topology optimization problems. The description of the method fully complies with any square, positive definite and symmetric system matrix. This is followed by a description of an example implementation and closed by discussions and conclusions.

## 2. Method

This section describes two different approaches to solve a generalized MPOP, namely (i) an elementary approach, and (ii) the novel work presented herein—the condensation approach. We start the description of the elementary approach of solving MPOPs, which will act as a reference.

### 2.1. Elementary approach

In the elementary approach one first defines the set of prescribed DOFs for each scenario. For each of the resulting sets of prescribed DOFs one can partition the system of linear equations accordingly. Subsequently, as in standard FEA one solves for the free state(s) [42].

The index set of all DOFs  $\mathbb{D}$  can be split in subset  $\mathbb{P}^{(i)} \subset \mathbb{D}$  containing the indices of all prescribed, and its complement  $\mathbb{F}^{(i)} := \mathbb{D} \setminus \mathbb{P}^{(i)}$  containing the indices of all free DOFs of analysis set  $i$ , hereafter denoted by subscripts 'p' and 'f', respectively. After partitioning accordingly, one may write the governing equations as

$$\begin{bmatrix} \mathbf{K}_{ff}^{(i)} & \mathbf{K}_{fp}^{(i)} \\ \mathbf{K}_{pf}^{(i)} & \mathbf{K}_{pp}^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_f^{(i)} \\ \mathbf{U}_p^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_f^{(i)} \\ \mathbf{F}_p^{(i)} \end{bmatrix}, \quad \forall i \in \mathbb{A}, \quad (7)$$

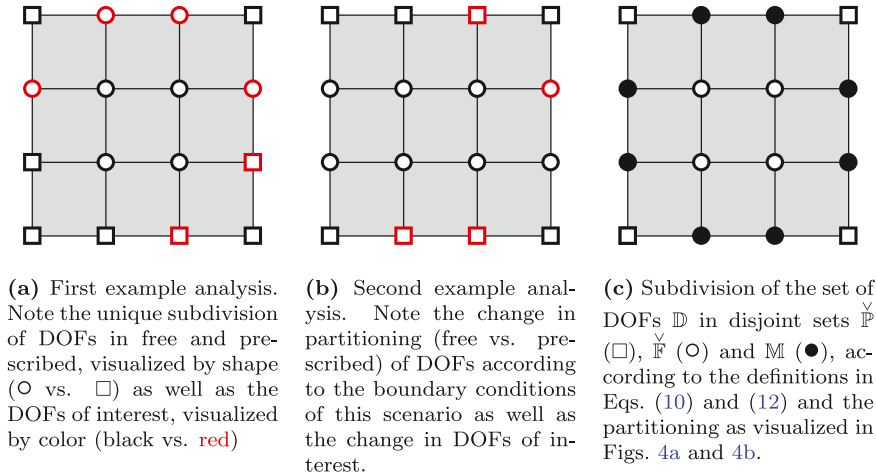
where  $\mathbf{U}_f^{(i)}$  are the free state(s),  $\mathbf{U}_p^{(i)}$  the prescribed state(s),  $\mathbf{F}_f^{(i)}$  the applied load(s) and  $\mathbf{F}_p^{(i)}$  the reaction load(s) of analysis set  $i$ . Note that  $\mathbf{U}_f^{(i)} \in \mathbb{R}^{f^{(i)} \times I^{(i)}}$ , where  $f^{(i)} := |\mathbb{F}^{(i)}|$ . Where subscript 'f' indicates the subset of the state and italic 'f' indicates the corresponding dimensionality of this set. The solutions to Eq. (7),  $\mathbf{U}_f^{(i)}$ , now can be obtained by solving the system of linear equations

$$\mathbf{K}_{ff}^{(i)} \mathbf{U}_f^{(i)} = \mathbf{F}_f^{(i)} - \mathbf{K}_{fp}^{(i)} \mathbf{U}_p^{(i)}, \quad \forall i \in \mathbb{A}. \quad (8)$$

Subsequently, if required, the reaction loads are obtained via

$$\mathbf{F}_p^{(i)} = \mathbf{K}_{pf}^{(i)} \mathbf{U}_f^{(i)} + \mathbf{K}_{pp}^{(i)} \mathbf{U}_p^{(i)}, \quad \forall i \in \mathbb{A}. \quad (9)$$





**Fig. 4.** Example of the discretized domain from Fig. 2(a) subjected to two distinct analyses; as shown in Figs. 4(a) and 4(b). The DOFs with prescribed states are denoted by a square ( $\square$ ), the DOFs with free states by a circle ( $\circ$ ) and primary DOFs by a black dot ( $\bullet$ ) (analogously to Figs. 2 and 3). DOFs of interest are coloured in red ( $\circ$ ,  $\square$ ). The combination of analyses leads to a subdivision in sets following Eqs. (10) and (12), as shown in Fig. 4(c). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Note that because of the uniqueness of each analysis set, following our definition,  $\mathbb{F}^{(i)} \neq \mathbb{F}^{(j)}$  and  $\mathbb{P}^{(i)} \neq \mathbb{P}^{(j)}$  for all  $i \neq j \in \mathbb{A}$ . Therefore, the number of times one has to preprocess a large system of equations equals  $a$ .

## 2.2. Condensation approach

We propose a novel method, aimed at solving MPOPs, consisting of two subsequent steps: condensation and analyses. First, one defines a set of primary DOFs that can fully describe the response function(s) and one condenses the remaining secondary DOFs to obtain a reduced system matrix. With this step, one basically solves for the secondary DOFs applying the same BCs for all analysis sets. Subsequently, a variety of analyses is performed based on this reduced system matrix with varying partitions, originating from the BCs of the scenarios. The latter step is fully equivalent to the elementary approach, however, as a result of the condensation, involves a much smaller system of linear equations. As such, one basically circumvents repeatedly solving for the secondary DOFs. Key is to define the set of primary DOFs in an adequate manner to omit repetitive analyses of large systems of equations, by restricting the size of the set of primary DOFs.

### 2.2.1. Definition of primary DOFs

Consider again the structure as visualized in Fig. 2(a). For the purpose of explanation, after discretization, we subject this arbitrary structure to two different scenarios as illustrated in Figs. 4(a) and 4(b). For each scenario one defines the DOFs with a prescribed state, indicated by a square in Fig. 4. In addition, one defines per scenario which DOFs are of interest for the response function of the MPOP, in Fig. 4 indicated in red. Note that both free and prescribed DOFs may be of interest. In what follows we explain how to find the smallest possible set of primary DOFs that can fully represent the behaviour of all scenarios and contains all DOFs of interest for a given MPOP.

For the condensation approach the index set of all DOFs  $\mathbb{D}$  is split in three disjoint parts, namely  $\mathbb{D} = \check{\mathbb{P}} \cup \check{\mathbb{F}} \cup \mathbb{M}$  as shown in Fig. 4(c). We define the proper index subset of prescribed secondary DOFs,  $\check{\mathbb{P}} \subsetneq \mathbb{D}$ , to consist of all indices of DOFs for which the state is prescribed in all analyses excluding those that are a DOF of interest for any of the analyses. Mathematically, this can be written as the intersection between all index sets of prescribed DOFs  $\mathbb{P}^{(i)}$  for which the DOFs are not a DOF of interest, *i.e.*

$$\check{\mathbb{P}} := \{x \in \{\cap_{i \in \mathbb{A}} \mathbb{P}^{(i)}\} \mid x \notin \{\cup_{i \in \mathbb{A}} \mathbb{M}^{(i)}\}\}. \quad (10)$$



Here  $\mathbb{M}^{(i)}$  consists of the indices of all DOFs of interest of analysis  $i$ . Fully analogously, the index set of free secondary DOFs,  $\check{\mathbb{F}} \subsetneq \mathbb{D}$ , is defined as all indices of DOFs for which the state is free in all analyses, excluding those that are a DOF of interest for any of the analyses. Based on these chosen definitions, the index set of primary DOFs,  $\mathbb{M}$ , is simply defined as the union between all indices of DOFs that are of interest in any of the analysis sets and the indices of DOFs that change freedom between analysis sets, *i.e.*

$$\mathbb{M} := (\cup_{i \in \mathbb{A}} \mathbb{M}^{(i)}) \cup ((\cup_{i \in \mathbb{A}} \mathbb{P}^{(i)}) \setminus (\cap_{i \in \mathbb{A}} \mathbb{P}^{(i)})). \quad (11)$$

Note this is also the complement of all secondary DOF indices, that is

$$\mathbb{M} := \mathbb{D} \setminus (\check{\mathbb{P}} \cup \check{\mathbb{F}}). \quad (12)$$

In many relevant applications,  $1 < m \ll n$ , meaning the number of primary DOFs is much lower than the total number of DOFs.

### 2.2.2. Static condensation

Considering a general MPOP, the response can now be rewritten as  $g[\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \dots, \hat{\mathbf{U}}^{(a)}]$ , with  $\hat{\mathbf{U}}^{(i)}$  the state of the primary DOFs of analysis set  $i$ . From the system matrix  $\mathbf{K} \in \mathbb{R}^{n \times n}$  one can obtain a reduced, but dense, system matrix  $\check{\mathbf{K}} \in \mathbb{R}^{m \times m}$  through static condensation, which exactly represents the system behaviour in terms of the primary DOFs. We can partition the system of linear equations in line with the subdivision in sets as proposed in Section 2.2.1, which gives

$$\begin{bmatrix} \check{\mathbf{K}} & \check{\mathbf{K}}_f & \check{\mathbf{K}}_p \\ \check{\mathbf{K}}_f & \check{\mathbf{K}}_{ff} & \check{\mathbf{K}}_{fp} \\ \check{\mathbf{K}}_p & \check{\mathbf{K}}_{pf} & \check{\mathbf{K}}_{pp} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}} \\ \check{\mathbf{U}}_f \\ \check{\mathbf{U}}_p \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}} \\ \check{\mathbf{F}}_f \\ \check{\mathbf{F}}_p \end{bmatrix}, \quad (13)$$

where  $\check{\mathbf{F}}_f \in \mathbb{R}^{\check{p} \times l}$  are the loads on the secondary DOFs and  $\check{\mathbf{U}}_p \in \mathbb{R}^{\check{p} \times l}$  the prescribed states of the secondary DOFs. Note that the primary states  $\hat{\mathbf{U}}$  contain both free and prescribed DOFs in accordance with the scenarios imposed, as will become evident from Section 2.2.3. Here we have defined  $l$  to be the sum of the loads in all analysis sets, *i.e.*

$$l := \sum_{i \in \mathbb{A}} l^{(i)}. \quad (14)$$

Hence, one can write for all  $i \in \mathbb{A}$

$$\check{\mathbf{F}}_f = [\check{\mathbf{F}}_f^{(1)}, \dots, \check{\mathbf{F}}_f^{(a)}], \quad \text{with} \quad \check{\mathbf{F}}_f^{(i)} = [\check{\mathbf{f}}_{f,1}^{(i)}, \dots, \check{\mathbf{f}}_{f,l^{(i)}}^{(i)}] \in \mathbb{R}^{\check{p} \times l^{(i)}}. \quad (15)$$

Analogously to Eq. (4), from the second row of Eq. (13), we obtain

$$\check{\mathbf{K}}_{ff} \check{\mathbf{U}}_f = \check{\mathbf{F}}_f - \check{\mathbf{K}}_f \hat{\mathbf{U}} - \check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p. \quad (16)$$

Substitution of Eq. (16) into the first row of Eq. (13) gives

$$\check{\mathbf{K}} \hat{\mathbf{U}} = \hat{\mathbf{F}} + \tilde{\mathbf{F}}. \quad (17)$$

Here  $\check{\mathbf{K}}$  is the reduced system matrix defined as

$$\check{\mathbf{K}} := \check{\mathbf{K}} - \check{\mathbf{K}}_f \check{\mathbf{X}}_f, \quad \check{\mathbf{K}} \in \mathbb{R}^{m \times m}, \quad (18)$$

and  $\tilde{\mathbf{F}}$  is the *reduced load*, defined as

$$\tilde{\mathbf{F}} := \check{\mathbf{K}}_f \check{\mathbf{V}}_f - \check{\mathbf{K}}_p \check{\mathbf{U}}_p, \quad \tilde{\mathbf{F}} \in \mathbb{R}^{m \times l}. \quad (19)$$

In order to obtain this reduced system of equations one has to solve the linear systems of equations

$$\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f = \check{\mathbf{K}}_f, \quad (20)$$

and

$$\check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f = \check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p - \check{\mathbf{F}}_f, \quad (21)$$

where we have introduced the ‘condensation states’  $\hat{\mathbf{X}}_f \in \mathbb{R}^{\check{f} \times m}$  and  $\check{\mathbf{V}}_f \in \mathbb{R}^{\check{f} \times l}$ . Note that the total number of analyses thus equals  $m + l$ . The method is thus capable of handling loads on the secondary DOFs, such as body loads. However, this will have an impact on the computational effort. The additional effort will—in practice—be modest, since (i) the analysis in Eqs. (20) and (21) belongs to the same analysis set, and (ii) the majority of variations in loading conditions will only affect the primary DOFs. As such, although  $l$  might be large, the number of unique loads will generally be a few.

The reduced system matrix  $\tilde{\mathbf{K}}$  consists of two parts:  $\hat{\mathbf{K}}$  containing the contributions of the primary DOFs, and  $\check{\mathbf{K}}_f \hat{\mathbf{X}}_f$  containing the contributions of the secondary DOFs reflected onto the primary DOFs. Moreover, by definition of Eq. (6),  $\tilde{\mathbf{K}}$  is square and symmetric. One should add the appropriate reduced loads to the loads on the primary DOFs in all subsequent analyses based upon the reduced system matrix. If the loads on the secondary DOFs  $\check{\mathbf{F}}_f = \mathbf{0}$  and the prescribed state on the secondary DOFs  $\check{\mathbf{U}}_p = \mathbf{0}$ , then the reduced load vanishes.

After solving Eq. (17), as will be described in the next section, one can retrieve the secondary states via

$$\check{\mathbf{U}}_f = -(\hat{\mathbf{X}}_f \hat{\mathbf{U}} + \check{\mathbf{V}}_f), \quad (22)$$

and subsequently the corresponding reaction loads on the secondary DOFs via

$$\check{\mathbf{F}}_p = \check{\mathbf{K}}_p \hat{\mathbf{U}} + \check{\mathbf{K}}_{pf} \check{\mathbf{U}}_f + \check{\mathbf{K}}_{pp} \check{\mathbf{U}}_p. \quad (23)$$

### 2.2.3. Analyses based on the reduced system matrix

After condensation, the reduced system matrix  $\tilde{\mathbf{K}}$  can be used to repeatedly analyse the reduced governing equations, and obtain the required states and/or reaction loads for the response(s) of the considered MPOP.

Reconsider the system of equations given in Eq. (17), i.e.

$$\tilde{\mathbf{K}} \hat{\mathbf{U}} = \hat{\mathbf{F}} + \tilde{\mathbf{F}}. \quad (24)$$

Note that the dimensionality of this system of equations is  $m$ . Similarly to the elementary approach, Section 2.1, one can split the indices in  $\mathbb{M}$  according to the type of boundary conditions, in the disjoint sets  $\mathbb{M} = \hat{\mathbb{F}}^{(i)} \sqcup \hat{\mathbb{P}}^{(i)}$  for each analysis set  $i$ . We define

$$\begin{aligned} \hat{\mathbb{F}}^{(i)} &:= \mathbb{M} \cap \mathbb{F}^{(i)}, \\ \hat{\mathbb{P}}^{(i)} &:= \mathbb{M} \cap \mathbb{P}^{(i)}. \end{aligned} \quad (25)$$

Partitioning of the system of linear equations accordingly gives

$$\begin{bmatrix} \tilde{\mathbf{K}}_{ff}^{(i)} & \tilde{\mathbf{K}}_{fp}^{(i)} \\ \tilde{\mathbf{K}}_{pf}^{(i)} & \tilde{\mathbf{K}}_{pp}^{(i)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_f^{(i)} \\ \hat{\mathbf{U}}_p^{(i)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_f^{(i)} \\ \hat{\mathbf{F}}_p^{(i)} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{F}}_f^{(i)} \\ \tilde{\mathbf{F}}_p^{(i)} \end{bmatrix}, \quad \forall i \in \mathbb{A}. \quad (26)$$

Note again, that this division of DOFs is unique for each set of analyses based upon  $\tilde{\mathbf{K}}$ , since  $\hat{\mathbb{F}}^{(i)} \neq \hat{\mathbb{F}}^{(j)}$  for all  $i \neq j \in \mathbb{A}$ . In addition to the partitioning, for each analysis set the appropriate loads are selected. Thus, whereas  $\tilde{\mathbf{F}} \in \mathbb{R}^{m \times l}$ ,  $\tilde{\mathbf{F}}_f^{(i)} \in \mathbb{R}^{\hat{f}^{(i)} \times l^{(i)}}$ . The subdivision is schematically visualized in Fig. 5, in line with the representation in Fig. 4.

The states of a variety of analyses sets can be obtained by solving the set of linear equations similar to the elementary approach, i.e.

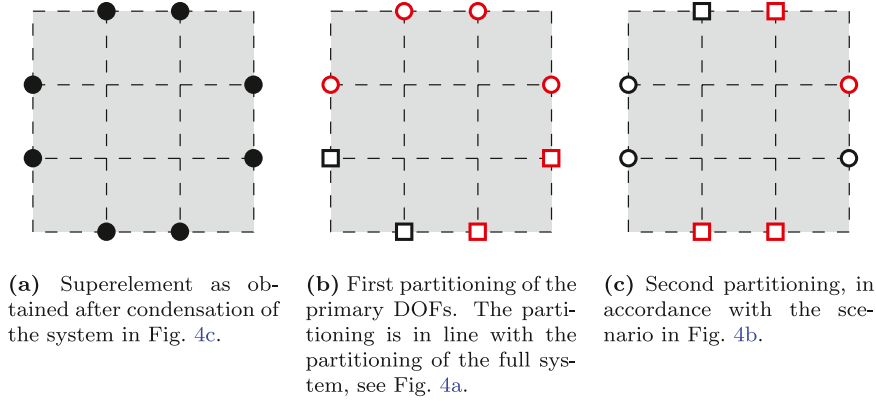
$$\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{U}}_f^{(i)} = \hat{\mathbf{F}}_f^{(i)} - \tilde{\mathbf{K}}_{fp}^{(i)} \hat{\mathbf{U}}_p^{(i)} + \tilde{\mathbf{F}}_f^{(i)}, \quad \forall i \in \mathbb{A}. \quad (27)$$

Note that system matrix  $\tilde{\mathbf{K}}_{ff}^{(i)}$  is square and non-singular for all  $i$ . If required, the reaction loads are obtained via

$$\hat{\mathbf{F}}_p^{(i)} = \tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{U}}_f^{(i)} + \tilde{\mathbf{K}}_{pp}^{(i)} \hat{\mathbf{U}}_p^{(i)} - \tilde{\mathbf{F}}_p^{(i)}, \quad \forall i \in \mathbb{A}. \quad (28)$$

### 2.3. Comparison of response evaluation

Although the approaches obtain *exactly* the same state and thereto response value(s), they differ in implementation, with critical consequences for the required computational effort. In order to keep an overview of the large



**Fig. 5.** Analyses on using the reduced system matrix, based on the example in Fig. 4. Fig. 5(a) shows the condensed domain of Fig. 4(c), resulting in a reduced system matrix. Figs. 5(b) and 5(c) illustrate the analyses based upon the reduced system matrix, in accordance with the scenarios of Figs. 4(a) and 4(b).

**Table 1**

Commonly used operators, dimensionality and first appearance equation number.

| Op                                      | Dim ( $\mathbb{R}$ )                   | Eqs. |
|---|--|------|
| $\mathbf{K}$                            | $n \times n$                           | (2)  |
| $\mathbf{K}_{\text{ff}}^{(i)}$          | $f^{(i)} \times f^{(i)}$               | (8)  |
| $\mathbf{U}_{\text{f}}^{(i)}$           | $f^{(i)} \times l^{(i)}$               | (8)  |
| $\tilde{\mathbf{K}}$                    | $m \times m$                           | (18) |
| $\tilde{\mathbf{F}}$                    | $m \times l$                           | (19) |
| $\tilde{\mathbf{K}}_{\text{ff}}^{\vee}$ | $\hat{f}^{\vee} \times \hat{f}^{\vee}$ | (20) |
| $\tilde{\mathbf{X}}_{\text{f}}^{\vee}$  | $\hat{f}^{\vee} \times m$              | (20) |
| $\tilde{\mathbf{V}}_{\text{f}}^{\vee}$  | $\hat{f}^{\vee} \times l$              | (20) |
| $\tilde{\mathbf{K}}_{\text{ff}}^{(i)}$  | $\hat{f}^{(i)} \times \hat{f}^{(i)}$   | (27) |
| $\tilde{\mathbf{U}}_{\text{f}}^{(i)}$   | $\hat{f}^{(i)} \times l^{(i)}$         | (27) |

number of sets, subscripts and different operators presented, the most important operators, their dimensionality and reference to their first use can be found in Table 1.

To highlight the differences, consider the pseudo codes in Algorithms 1 and 2, that describe the procedures to obtain the same response function using the elementary and condensation approach, respectively. Note the *critical* difference between the solves in Algorithm 1: Line 2 and Algorithm 2: Line 10. To obtain the state(s) of the DOFs of interest, in the condensation approach, one solves for the state using the partitioned reduced system matrices  $\tilde{\mathbf{K}}_{\text{ff}}^{(i)} \in \mathbb{R}^{\hat{f}^{(i)} \times \hat{f}^{(i)}}$ , where the number of free primary DOFs  $\hat{f}^{(i)} := m - \hat{p}^{(i)}$  and  $\hat{p}^{(i)} := |\hat{\mathbb{P}}^{(i)}|$ . In the elementary approach, one solves for the state using system matrices  $\mathbf{K}_{\text{ff}}^{(i)} \in \mathbb{R}^{f^{(i)} \times f^{(i)}}$ , where the number of free DOFs  $f^{(i)} := n - p^{(i)}$  and  $p^{(i)} := |\mathbb{P}^{(i)}|$ . Since  $f^{(i)} \approx n$  and  $\hat{f}^{(i)} \approx m$ , for sufficiently large systems, the dimensionality of the constrained system matrix  $\mathbf{K}_{\text{ff}}^{(i)}$  is much larger than the constrained reduced system matrix  $\tilde{\mathbf{K}}_{\text{ff}}^{(i)}$ , that is  $n \gg m$  for all analysis sets  $i$ . The difference in computational effort of preprocessing a system matrix of dimensionality  $n \times n$  as compared to  $m \times m$  is large. The condensation approach only requires a single preprocessing of a large system matrix (Algorithm 2: Line 4), compared to a preprocessing steps of a large system matrix for the elementary approach (Algorithm 1: Line 2). As a result, the computational effort to evaluate the response(s) can be highly reduced by using the condensation approach, when multiple scenarios are considered.

Upon close examination of the approaches, it becomes apparent the elementary approach contains the differences between the BCs of analysis sets in the large system of equations (Algorithm 1: Line 2), effectively solving *repeatedly* for the large amount of secondary DOFs. In contrast, the condensation approach basically uses some

pre-processing (the static condensation in Algorithm 2: Line 4) to solve for the secondary DOFs. The different partitionings are ‘moved’ to the system of equations of reduced dimensionality, see Algorithm 2: Line 10.

In terms of implementation, little is required to extend the elementary approach to the condensation approach. Apart from additional ‘administrative tasks’ involving index sets, *i.e.* Algorithm 2: Lines 1–3 and 8–9, the extension requires additional matrix–vector multiplications (Lines 5–6) and solves (Lines 4 and 10). Irrespective of the programming language of choice, those operations are also required for the elementary approach. The similarity between approaches in terms of implementation becomes all the more evident upon examination of the attached MATLAB code.

---

**Algorithm 1** elementary approach: response evaluation
 

---

```

1: for all  $i \in \mathbb{A}$  do
2:   solve  $\mathbf{K}_{\text{ff}}^{(i)} \mathbf{U}_f^{(i)} = \mathbf{F}_f^{(i)} - \mathbf{K}_{\text{fp}}^{(i)} \mathbf{U}_p^{(i)}, \quad \mathbf{K}_{\text{ff}}^{(i)} \in \mathbb{R}^{f^{(i)} \times f^{(i)}}$ 
3: end for
4: return  $g \leftarrow g[\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(a)}]$ 

```

---



---

**Algorithm 2** Condensation approach: response evaluation
 

---

```

1:  $\check{\mathbb{P}} := \{x \in \{\cap_{i \in \mathbb{A}} \mathbb{P}^{(i)}\} \mid x \notin \{\cup_{i \in \mathbb{A}} \mathbb{M}^{(i)}\}\}$ 
2:  $\check{\mathbb{F}} := \{x \in \{\cap_{i \in \mathbb{A}} \mathbb{F}^{(i)}\} \mid x \notin \{\cup_{i \in \mathbb{A}} \mathbb{M}^{(i)}\}\}$ 
3:  $\mathbb{M} \leftarrow \mathbb{D} \setminus (\check{\mathbb{P}} \cup \check{\mathbb{F}})$ 
4: solve  $\check{\mathbf{K}}_{\text{ff}} \begin{bmatrix} \check{\mathbf{X}}_f \\ \check{\mathbf{V}}_f \end{bmatrix} = \begin{bmatrix} \check{\mathbf{K}}_f & \check{\mathbf{K}}_{\text{fp}} \end{bmatrix} \check{\mathbf{U}}_p - \check{\mathbf{F}}_f, \quad \check{\mathbf{K}}_{\text{ff}} \in \mathbb{R}^{\check{f} \times \check{f}}$ 
5:  $\tilde{\mathbf{K}} \leftarrow \check{\mathbf{K}} - \check{\mathbf{K}}_f \check{\mathbf{X}}_f$ 
6:  $\tilde{\mathbf{F}} \leftarrow \check{\mathbf{K}}_f \check{\mathbf{V}}_f - \check{\mathbf{K}}_p \check{\mathbf{U}}_p$ 
7: for all  $i \in \mathbb{A}$  do
8:    $\hat{\mathbb{F}}^{(i)} \leftarrow \mathbb{M} \cap \mathbb{F}^{(i)}$ 
9:    $\hat{\mathbb{P}}^{(i)} \leftarrow \mathbb{M} \cap \mathbb{P}^{(i)}$ 
10:  solve  $\tilde{\mathbf{K}}_{\text{ff}}^{(i)} \hat{\mathbf{U}}_f^{(i)} = \hat{\mathbf{F}}_f^{(i)} - \tilde{\mathbf{K}}_{\text{fp}}^{(i)} \hat{\mathbf{U}}_p^{(i)} + \tilde{\mathbf{F}}_f^{(i)}, \quad \tilde{\mathbf{K}}_{\text{ff}}^{(i)} \in \mathbb{R}^{\hat{f}^{(i)} \times \hat{f}^{(i)}}$ 
11: end for
12: return  $g \leftarrow g[\hat{\mathbf{U}}^{(1)}, \dots, \hat{\mathbf{U}}^{(a)}]$ 

```

---

### 3. Sensitivity analysis

Design sensitivities are essential for gradient-based structural optimization schemes, as they provide the basis for the design update. This section describes the sensitivity analysis of all relevant entries used in the condensation approach, as described in Section 2.2. Since the second part of the condensation approach, as described in Section 2.2.3, also covers the elementary approach we omit a separate description of the sensitivities for the elementary approach.

Making use of the chain rule of differentiation, the full derivative of a response function  $g[\hat{\mathbf{U}}[\tilde{\mathbf{K}}[\mathbf{K}[\mathbf{x}]], \tilde{\mathbf{F}}[\mathbf{K}[\mathbf{x}]]]$  to design variable  $x_k$  reads as

$$\frac{dg}{dx_k} = \frac{dg}{d\hat{\mathbf{U}}} : \left( \frac{\partial \hat{\mathbf{U}}}{\partial \tilde{\mathbf{K}}} : \frac{\partial \tilde{\mathbf{K}}}{\partial \mathbf{K}} + \frac{\partial \hat{\mathbf{U}}}{\partial \tilde{\mathbf{F}}} : \frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{K}} \right) : \frac{d\mathbf{K}}{dx_k}. \quad (29)$$

Herein, we assume  $\frac{dg}{d\hat{\mathbf{U}}}$  and  $\frac{d\mathbf{K}}{dx_k}$  are known, since they are problem and response dependent. Although not a restriction of the method, for simplicity of the discussion in this section, we assume design independent loads. This section is subdivided into the separate sensitivity analysis of

1. the reduced system matrix  $\tilde{\mathbf{K}}$  to the system matrix  $\mathbf{K}$ ,
2. the reduced load(s)  $\tilde{\mathbf{F}}$  to the system matrix  $\mathbf{K}$ , and
3. the primary state(s)  $\hat{\mathbf{U}}$  to the reduced system matrix  $\tilde{\mathbf{K}}$  and load(s)  $\tilde{\mathbf{F}}$ .

The remaining sensitivity analyses, *e.g.* the sensitivities of the reaction loads  $\hat{\mathbf{F}}_p$  with respect to the reduced system matrix, are not considered critical for understanding of the method and subsequent conclusions. However, for completeness those and additional sensitivities can be found in [Appendix A](#).

In the following, the sub-matrices of the system matrices in Eqs. (13) and (26) are written using selection matrices. Some examples are:

$$\begin{aligned}\mathbf{K}_{pf} &:= \mathbf{S}_p^T \mathbf{K} \mathbf{S}_f \\ \hat{\hat{\mathbf{K}}} &:= \hat{\mathbf{S}}^T \hat{\mathbf{K}} \hat{\mathbf{S}} \\ \check{\mathbf{K}}_f &:= \check{\mathbf{S}}^T \check{\mathbf{K}} \check{\mathbf{S}}_f \\ \tilde{\mathbf{K}}_{pf} &:= \tilde{\mathbf{S}}_p^T \tilde{\mathbf{K}} \tilde{\mathbf{S}}_f\end{aligned}\tag{30}$$

These highly sparse rectangular selection matrices  $\mathbf{S}_{(\cdot)}^{(\cdot)}$  can be used to select parts from the system matrix, which will prove useful during the sensitivity analysis.

### 3.1. Sensitivities of the reduced system matrix

Consider a response  $g \left[ \hat{\mathbf{U}} \left[ \tilde{\mathbf{K}} [\mathbf{K} [\mathbf{x}]] \right] \right]$ , with

$$\tilde{\mathbf{K}} := \hat{\hat{\mathbf{K}}} - \check{\mathbf{K}}_f \check{\mathbf{X}}_f.\tag{18 revisited}$$

Here the system matrix is a function of design variables  $\mathbf{x}$ , *i.e.*  $\mathbf{K} [\mathbf{x}]$ . However, for demonstration of the method we are solely interested in the sensitivities of the reduced system matrix  $\tilde{\mathbf{K}}$  with respect to the system matrix  $\mathbf{K}$ . Therefore, this dependency is omitted from here onward. For conciseness, we use the full differential. The full derivative of the response function can be written as

$$dg[\tilde{\mathbf{K}}[\mathbf{K}]] = \frac{dg}{d\tilde{\mathbf{K}}} : \frac{d\tilde{\mathbf{K}}}{d\mathbf{K}} : d\mathbf{K},\tag{31}$$

where  $\frac{dg}{d\tilde{\mathbf{K}}} = \frac{dg}{d\hat{\mathbf{U}}} : \frac{d\hat{\mathbf{U}}}{d\tilde{\mathbf{K}}}$  is the sensitivity of the response with respect to  $\tilde{\mathbf{K}}$  and  $d\mathbf{K}$  the sensitivity of the system matrix.

In the following we assume both are known. The aim is to find  $\frac{d\tilde{\mathbf{K}}}{d\mathbf{K}}$ , *i.e.* the sensitivities of the reduced system matrix with respect to the system matrix.

The total derivative of  $\tilde{\mathbf{K}}$  is given by

$$d\tilde{\mathbf{K}}[\mathbf{K}] = d\hat{\hat{\mathbf{K}}} - d\check{\mathbf{K}}_f \check{\mathbf{X}}_f + \check{\mathbf{X}}_f d\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f - \check{\mathbf{X}}_f d\check{\mathbf{K}}_f.\tag{32}$$

Here we have made use of

$$d\check{\mathbf{X}}_f = -\check{\mathbf{K}}_{ff}^{-1} d\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f + \check{\mathbf{K}}_{ff}^{-1} d\check{\mathbf{K}}_f.\tag{33}$$

Substitution of Eq. (30) into Eq. (32) yields

$$d\tilde{\mathbf{K}}[\mathbf{K}] = \left( \hat{\mathbf{S}} - \check{\mathbf{S}}_f \check{\mathbf{X}}_f \right)^T d\mathbf{K} \left( \hat{\mathbf{S}} - \check{\mathbf{S}}_f \check{\mathbf{X}}_f \right).\tag{34}$$

By rewriting<sup>2,3</sup> Eq. (34) we obtain the full derivative

$$d\tilde{\mathbf{K}}[\mathbf{K}] = (\mathbf{A} \otimes \mathbf{A}) : d\mathbf{K},\tag{35}$$

with

$$\mathbf{A} := \hat{\mathbf{S}} - \check{\mathbf{S}}_f \check{\mathbf{X}}_f.\tag{36}$$

<sup>2</sup> Here we introduce the dyadic product, also known and equivalent to the outer or tensor product. The dyadic product of two vectors  $\mathbf{a} \in \mathbb{R}^n$  and  $\mathbf{b} \in \mathbb{R}^m$  can be represented as  $\mathbf{a} \otimes \mathbf{b} \equiv \mathbf{a}\mathbf{b}^T \equiv a_i b_j$ . For matrices  $\mathbf{A} \in \mathbb{R}^{n \times c}$  and  $\mathbf{B} \in \mathbb{R}^{m \times c}$ , the dyadic product is defined as  $\mathbf{A} \otimes \mathbf{B} \equiv \mathbf{A}\mathbf{B}^T \equiv A_{ij} B_{kj}$ .

<sup>3</sup>  $\mathbf{A}^T \mathbf{B} \mathbf{C} \equiv A_{ij} B_{jk} C_{ki} = A_{ji} C_{ik} B_{jk} = (\mathbf{A}\mathbf{C}^T) : \mathbf{B} \equiv (\mathbf{A} \otimes \mathbf{C}) : \mathbf{B}$ .

Thus

$$\frac{d\tilde{\mathbf{K}}}{d\mathbf{K}} = \mathbf{A} \otimes \mathbf{A}, \quad (37)$$

which consists only of terms that have been previously computed in the condensation. The following section further explores this property of the reduced system matrix. The full step-by-step derivation, both for the direct and adjoint approach, can be found in [Appendices A.1.1](#) and [A.1.2](#).

### 3.1.1. On the self-adjointness of the reduced system matrix

Following the adjoint sensitivity approach [43,44] we define

$$\mathcal{L}[\tilde{\mathbf{K}}[\mathbf{K}], \overset{\vee}{\Lambda}_f] = g[\tilde{\mathbf{K}}[\mathbf{K}]] + \overset{\vee}{\Lambda}_f : \left( \overset{\vee}{\mathbf{K}}_{ff} \overset{\vee}{\mathbf{X}}_f - \overset{\vee}{\mathbf{K}}_f \right), \quad (38)$$

where  $\overset{\vee}{\Lambda}_f$  are the Lagrange multipliers. The sensitivities of the Lagrangian (extensive derivation can be found in [Appendix A.1.2](#)) are obtained via

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{K}}} : \left( \overset{\vee}{\mathbf{S}}^T d\mathbf{K} \mathbf{A} \right) + \overset{\vee}{\Lambda}_f : \left( \overset{\vee}{\mathbf{S}}_f^T d\mathbf{K} \mathbf{A} \right). \quad (39)$$

To find  $\overset{\vee}{\Lambda}_f$ , one has to solve the following system of equations:

$$\overset{\vee}{\mathbf{K}}_{ff} \overset{\vee}{\Lambda}_f = \overset{\vee}{\mathbf{K}}_f \frac{dg}{d\tilde{\mathbf{K}}}. \quad (40)$$

By suitable choice of calculation order [45], this can be rewritten as

$$\overset{\vee}{\Lambda}_f = \overset{\vee}{\mathbf{K}}_{ff}^{-1} \left( \overset{\vee}{\mathbf{K}}_f \frac{dg}{d\tilde{\mathbf{K}}} \right) = \overset{\vee}{\mathbf{X}}_f \frac{dg}{d\tilde{\mathbf{K}}}. \quad (41)$$

In contrast to the general adjoint approach, where the calculation order is determined by the number of responses versus the number of design variables, the most suitable choice in this case is the same under all conditions, since  $\overset{\vee}{\mathbf{X}}_f$  has already been calculated. Thus, the sensitivities of the reduced system matrix can be calculated without solving an additional system of linear equations. Since both the state and adjoint fields are exactly identical we can conclude that—for a symmetric system matrix—an optimization problem with response  $g[\tilde{\mathbf{K}}]$  has a *self-adjoint* operator [5,44] independent of the self-adjointness of the operator(s)  $\tilde{\mathbf{K}}$  is acting upon. As a result, no extra preprocessing is required to compute the sensitivities of  $\tilde{\mathbf{K}}$ . This has critical consequences with respect to the computational efficiency of the condensation approach.

### 3.2. Sensitivities of the reduced load

Since the required steps to obtain the sensitivities of the reduced load are very similar to those in the previous section we solely present the result here. The full derivation—both for the direct and adjoint approach—can be found in [Appendices A.2.1](#) and [A.2.2](#).

Recall the definition of the reduced load, namely

$$\tilde{\mathbf{F}}[\mathbf{K}, \overset{\vee}{\mathbf{F}}_f, \overset{\vee}{\mathbf{U}}_p] := \tilde{\mathbf{K}}_f \overset{\vee}{\mathbf{V}}_f - \tilde{\mathbf{K}}_p \overset{\vee}{\mathbf{U}}_p. \quad (19 \text{ revisited})$$

Let us consider a response function  $g[\tilde{\mathbf{F}}[\mathbf{K}]]$ , hence omitting the dependency on the free secondary load(s)  $\overset{\vee}{\mathbf{F}}_f$  and prescribed secondary state(s)  $\overset{\vee}{\mathbf{U}}_p$  here. Substitution of Eq. (30) into Eq. (19) and application of either direct or adjoint sensitivity analysis yields

$$\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{K}} = \mathbf{A} \otimes \mathbf{B} \quad (42)$$

with

$$\mathbf{B} := \overset{\vee}{\mathbf{S}}_f \overset{\vee}{\mathbf{V}}_f - \overset{\vee}{\mathbf{S}}_p \overset{\vee}{\mathbf{U}}_p. \quad (43)$$

Note that Eq. (42) solely involves quantities that have been calculated in the condensation process and requires therefore negligible additional computational effort.

### 3.3. Sensitivities of the state

In an MPOP the response function, often  $g \left[ \hat{\mathbf{U}}_f \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}}, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$ , generally depends on multiple states, *i.e.*  $g \left[ \hat{\mathbf{U}}_f^{(1)}, \dots, \hat{\mathbf{U}}_f^{(a)} \right]$  is obtained by solving

$$\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{U}}_f^{(i)} = \hat{\mathbf{F}}_f^{(i)} - \tilde{\mathbf{K}}_{fp}^{(i)} \hat{\mathbf{U}}_p^{(i)} + \tilde{\mathbf{F}}_f^{(i)}, \quad \forall i \in \mathbb{A}. \quad (27 \text{ revisited})$$

Therefore, one is required to determine the total derivative  $dg \left[ \hat{\mathbf{U}}_f^{(i)} \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}}^{(i)}, \hat{\mathbf{F}}_f^{(i)}, \hat{\mathbf{U}}_p^{(i)} \right] \right]$  for all  $i \in \mathbb{A}$ . For clarity, the superscript  $i$  is omitted from here on. Moreover, for conciseness we omit the dependency on the free loads  $\hat{\mathbf{F}}_f$  and prescribed states  $\hat{\mathbf{U}}_p$  in this section. Although this section focuses on the condensation approach, the following is similar for the elementary approach, where  $g \left[ \mathbf{U}_f \left[ \mathbf{K} \right] \right]$ . Complete and step-by-step derivations can be found in [Appendix A.3](#).

The full derivative of the response function is given by

$$dg \left[ \hat{\mathbf{U}}_f \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}} \right] \right] = \frac{dg}{d\hat{\mathbf{U}}_f} : \left( \frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{K}}} : d\tilde{\mathbf{K}} + \frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{F}}} : d\tilde{\mathbf{F}} \right). \quad (44)$$

We aim to find  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{K}}}$  and  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{F}}}$ , while assuming  $d\tilde{\mathbf{K}}$ ,  $d\tilde{\mathbf{F}}$  and  $\frac{dg}{d\hat{\mathbf{U}}_f}$  to be known. The full derivative  $d\hat{\mathbf{U}}_f$  can be obtained by differentiation of Eq. (27), which gives

$$d\hat{\mathbf{U}}_f \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}} \right] = \tilde{\mathbf{K}}_{ff}^{-1} \left( d\tilde{\mathbf{F}}_f - d\tilde{\mathbf{K}}_{ff} \hat{\mathbf{U}}_f - d\tilde{\mathbf{K}}_{fp} \hat{\mathbf{U}}_p \right). \quad (45)$$

Substitution of Eq. (30) and simplification yields

$$\frac{\partial g}{\partial \tilde{\mathbf{K}}} = -\tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f \otimes \hat{\mathbf{U}} \quad (46)$$

and

$$\frac{\partial g}{\partial \tilde{\mathbf{F}}} = \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f, \quad (47)$$

where

$$\tilde{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f = \frac{dg}{d\hat{\mathbf{U}}_f} \quad (48)$$

is the adjoint equation and  $\hat{\mathbf{A}}^{(i)} \in \mathbb{R}^{m \times l^{(i)}}$  is the state of the adjoint problem of analysis set  $i$ . Thus, to obtain  $d\hat{\mathbf{U}}_f \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}} \right]$  one requires one additional system of linear equations to be solved *per analysis set*.

The total number of adjoint loads to solve an MPOP theoretically equals the number of responses times the number of states, that is the total number of adjoint loads

$$b = hl = h \sum_{i=1}^a l^{(i)}, \quad (49)$$

with  $h$  the number of responses of the MPOP. However, if the right-hand-side is linear dependent on the applied load, the problem is self-adjoint and solving the corresponding adjoint equation can be omitted [5]. This is for example the case in classical compliance minimization [46]. What is more, in practice, the majority of responses depend on a single state or multiple states of a single analysis set, that is many adjoint loads  $\frac{dg_k}{du_j^{(i)}}$  are zero. Thereto, generally the total number of adjoint loads satisfies

$$b = \sum_{j=1}^h \sum_{i=1}^a b_j^{(i)}, \quad (50)$$

with  $b_j^{(i)}$  the number of non-zero adjoint loads of response function  $j$  to the states of analysis set  $i$ , and thus  $b^{(i)} \leq l^{(i)}$ .



### 3.4. Comparison of sensitivity analysis

To compare the sensitivity analyses, consider the pseudo-codes Algorithms 3 and 4, respectively for the elementary and condensation approach. Herein, it is assumed both applied loads and prescribed states are design independent. Note the difference in dimensionality of the systems of equations to be solved in the second step of the algorithms, that is Algorithm 3: Line 2 and Algorithm 4: Line 2. It may be assumed here no additional preprocessing is required, as the preprocessing of the response evaluation can be reused. As a result, in the elementary approach, for each response  $g_j$ ,  $b_j$  adjoint states of substantial dimensionality are to be solved for. In contrast, assuming  $m \ll n$ , the computational effort of the sensitivity analysis in the condensation approach is negligible. Most notably is the fact that *zero* solves of substantial dimensionality are required, see Section 3.1.1.

Apart from some additional matrix–vector operations, see Algorithm 4: Lines 4–5, implementation of the sensitivity analysis of the condensation approach is straightforward. Also note the similarity between the structure of Algorithm 3: Line 4 and Algorithm 4: Line 6.

Full step-by-step sensitivity analysis of  $g \left[ \hat{\mathbf{U}}_f \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$  and  $g \left[ \hat{\mathbf{F}}_p \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$ , without those assumptions, can be found in Appendices A.3 and A.4. In addition, the derivation of sensitivities for responses involving the state or reaction load of secondary DOFs, that is  $g \left[ \check{\mathbf{U}}_f \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$  and  $g \left[ \check{\mathbf{F}}_p \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$ , can be found in Appendices A.5 and A.6. All resulting sensitivities are summarized in Table A.3 of Appendix A.7.

---

#### Algorithm 3 elementary approach: sensitivity analysis

---

```

1: for all  $i \in \mathbb{A}$  do
2:   solve  $\mathbf{K}_{\text{ff}}^{(i)} \mathbf{A}_f^{(i)} = \frac{\partial g}{\partial \mathbf{U}_f^{(i)}}$ ,  $\mathbf{K}_{\text{ff}}^{(i)} \in \mathbb{R}^{f^{(i)} \times f^{(i)}}$ 
3: end for
4: return  $\frac{\partial g}{\partial x_k} \leftarrow - \sum_{i \in \mathbb{A}} \sum_{j=1}^{l^{(i)}} \lambda_j^{(i)} \cdot \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}_j^{(i)}$ 

```

---



---

#### Algorithm 4 Condensation approach: sensitivity analysis

---

```

1: for all  $i \in \mathbb{A}$  do
2:   solve  $\tilde{\mathbf{K}}_{\text{ff}}^{(i)} \hat{\mathbf{A}}_f^{(i)} = \frac{\partial g}{\partial \hat{\mathbf{U}}_f^{(i)}}$ ,  $\tilde{\mathbf{K}}_{\text{ff}}^{(i)} \in \mathbb{R}^{\hat{f}^{(i)} \times \hat{f}^{(i)}}$ 
3: end for
4:  $\mathbf{A} \leftarrow \hat{\mathbf{S}} - \check{\mathbf{S}}_f \check{\mathbf{X}}_f$ 
5:  $\mathbf{B} \leftarrow \check{\mathbf{S}}_f \check{\mathbf{V}}_f - \check{\mathbf{S}}_p \check{\mathbf{U}}_p$ 
6: return  $\frac{\partial g}{\partial x_k} \leftarrow \sum_{i \in \mathbb{A}} \sum_{j=1}^{l^{(i)}} \left( \mathbf{A} \hat{\lambda}_j^{(i)} \right) \cdot \frac{\partial \mathbf{K}}{\partial x_k} \left( \mathbf{b}_j^{(i)} - \mathbf{A} \hat{\mathbf{u}}_j^{(i)} \right)$ 

```

---

## 4. Computational effort

This section describes the computational effort involved in using the condensation approach compared to the elementary approach. First, predominant factors that influence the computational effort are discussed. Subsequently, the computational effort of both approaches is estimated using algorithmic complexity analysis, which yields an estimate of the computational efficiency of using the condensation approach with respect to the elementary approach as a function of the predominant factors.

### 4.1. Predominant factors

A critical factor influencing the computational effort of the condensation approach is the number of free secondary DOFs versus number of primary DOFs. The number of free secondary DOFs  $\check{f}$  depends on the system size  $n$ , the number of prescribed secondary DOFs  $\check{p}$  and the number of primary DOFs  $m$  via  $\check{f} = n - \check{p} - m$ . The number

of free secondary DOFs  $\check{f}$  is the primary contributor to the total effort of calculating the Schur complement, as it directly influences the cost of the preprocessing of  $\check{\mathbf{K}}_{\text{ff}}$ . The number of primary DOFs  $m$  determines the number of right-hand-sides of the system of linear equations solved in the static condensation process. If the number of primary DOFs increases, the number of secondary DOFs decreases with equal amount. In addition, the number of primary DOFs  $m$  influences the computational effort of the subsequent analysis sets based upon  $\check{\mathbf{K}}$ . The computational effort of subsequent analysis sets increases rapidly as  $m$  increases, since  $\check{\mathbf{K}}$  is a dense system matrix as compared to the generally sparse system matrix  $\mathbf{K}$ .

The similarity between analysis sets or, equivalently, the amount of overlap of sets  $\mathbb{F}^{(i)}$  for all analysis sets is crucial. The higher the overlap of those sets, the smaller  $\mathbb{M}$  can be, and hence, the lower becomes the computational effort of both the static condensation and subsequent analysis sets. The presence of non-zero prescribed values on the secondary DOFs, i.e.  $\check{\mathbf{U}}_{\text{p}} \neq \mathbf{0}$  and/or non-zero applied loads on the secondary DOFs  $\check{\mathbf{F}}_{\text{f}} \neq \mathbf{0}$  introduces  $l$  extra right-hand-sides for the static condensation and introduces a reduced load term. In that case, the number of prescribed secondary DOFs  $\check{p}$  influences the effort of matrix–vector product  $\check{\mathbf{K}}_{\text{fp}}\check{\mathbf{U}}_{\text{p}}$  and  $\check{\mathbf{K}}_{\text{p}}\check{\mathbf{U}}_{\text{p}}$  if  $\check{\mathbf{U}}_{\text{p}} \neq \mathbf{0}$ , and influences  $\check{f}$  via  $\check{f} = n - m - \check{p}$ . However, for the majority of large scale problems  $\check{p} \ll n$  and the additional effort becomes negligible.

Another clearly important factor influencing the computational effort is the number of analysis sets  $a$ . The computational effort of the elementary approach increases linearly with  $a$ . In contrast, in the condensation approach  $a$  has little influence on the computational effort if  $m \ll n$ , as analysis sets based upon  $\check{\mathbf{K}}$  are—in that case—of negligible effort. In contrast, when  $m \rightarrow n$  both the static condensation and all analysis sets based upon  $\check{\mathbf{K}}$  become expensive, due to the high number of right-hand-sides in the static condensation and the large dense matrix  $\check{\mathbf{K}}$ . For the elementary approach, most important is the number of free DOFs for each analysis set  $f^{(i)}$ , which determines the effort of the preprocessing of  $\mathbf{K}_{\text{ff}}^{(i)}$ . Note that generally  $f^{(i)} \approx f^{(j)}$  for all  $i, j \in \mathbb{A}$ . The number of loads  $l^{(i)}$  for each analysis set will have a positive influence on the efficiency of the condensation approach with respect to the elementary approach for  $m \ll n$ , but *vice versa* if  $m \rightarrow n$ .

Also the sparsity, condition number, bandedness and band structure of  $\mathbf{K}$  influence the type and cost of preprocessing and, hence, to a lesser extent impact the computational effort of both approaches. Such properties mostly depend on the type of governing equations, but are also influenced by the discretization (e.g. node numbering and reordering) and dimension (e.g. band structure and sparsity are different for 2D and 3D problems). Lastly, implementation-related factors play a role, e.g. type of solver, compiled vs. interpreted code, vectorization and parallelization. Note that some of the dominant parameters mostly affect the computational effort of the condensation approach (such as the number of analyses  $a$  or dimensionality of  $\check{\mathbf{K}}$ ), whereas others will have a comparable influence on both approaches, e.g. the type of solver or specific implementation.

#### 4.2. Prognosis

Based on the previous discussion we expect the computational efficiency of the condensation approach with respect to the elementary approach to be high when  $m \ll n$  and to increase linearly with  $a$ . Assuming  $m \ll n$  and  $a \geq 1$ , we expect the source of computational efficiency of the condensation approach with respect to the elementary approach to be twofold:

1. The condensation approach requires a single preprocessing of the system matrix, as opposed to  $a$  for the elementary approach.<sup>4</sup> All subsequent analysis sets based on the reduced system matrix have relatively negligible computational effort. Thus, the more analysis sets are based upon  $\check{\mathbf{K}}$ , the higher will be the computational efficiency with respect to the elementary approach.
2. As described in Section 3.1.1, the sensitivities of the reduced system matrix are self-adjoint, independent of the response. All information required to compute those sensitivities has been calculated during the condensation process. What remains is the computational effort of the adjoint problems based on the reduced system matrix. The problem may still consist of multiple non-self-adjoint responses, but when  $m \ll n$ , the associated adjoint problems are negligibly small compared to the condensation process required for the

<sup>4</sup> Note that in the elementary approach, if using a direct solution method, one also has to store  $a$  system matrices of notable size in memory, as opposed to a single matrix in the condensation approach.

response evaluation. The sensitivity analysis of any response depending solely on the reduced system matrix is therefore of negligible computational effort.

For a fixed number of analysis sets  $a$ , an increase in  $m$  causes an increase in computational efficiency due to the low cost of repetitive analysis sets based upon the reduced system matrix in contrast to the high cost of each extra analysis set in the elementary approach. However, when  $m$  becomes large, the static condensation becomes computationally more demanding. Of even more importance is the increase in dimensionality of the reduced system matrix, which is a dense matrix. Based on the above, it is expected that the gain is higher than one (meaning computational effort of condensation approach is less than the effort of the elementary approach) for any problem with more than one analysis  $a$  and any ratio of  $\eta := \frac{m}{n}$  that satisfies  $0 < \underline{\eta} \leq \eta \leq \bar{\eta} < 1$ , with  $\underline{\eta}$  and  $\bar{\eta}$  the bounds on  $\eta$  for which positive gain is expected.

#### 4.3. Algorithmic complexity analysis

To fairly compare the computational efficiency of the condensation approach to the elementary approach, we investigate the computational gain  $\Xi$ , defined as the relative decrease in computational effort through the use of the condensation approach compared to the elementary approach, as a function of the predominant parameters. We define the theoretical computational effort as the number of floating point operations (FLOPs)  $\beta$  required and assume that the relative FLOP count of the two algorithms is an approximation that gives sufficient accuracy to estimate the resulting relative Central Processing Unit (CPU) time  $t$ . In what follows it will be shown that this assumption is sufficiently valid under the chosen conditions to get a simple order of magnitude estimator. The computational gain can be based upon the theoretical estimated FLOP count, denoted by  $\Xi_{\tilde{\beta}}$  or, alternatively, based on experimentally measured run time, hereafter denoted by  $\Xi_t$ . First, this section presents an algorithmic complexity analysis, followed by a numerical study in Section 5. The overset of  $\tilde{\beta}$  with a tilde emphasizes, in contrast to the exact FLOP count  $\beta$ , that the FLOP count is an estimation. The goal of the theoretical estimation is to have a ‘back-of-the-envelope’ measure of the *order of magnitude gain one may expect*. Considering the number and severity of the assumptions that will be required to get a reasonably simple analytical formula for the estimation, the resulting formulae are by no means expected to be an exact computational effort predictor.

Note that computational gain is measured in relative FLOPs or relative runtime, whereas—in some cases—storage considerations might be critical. A thorough investigation of memory storage (as well as parallelization) of the method is considered out of scope of the present work, nonetheless some critical observations can be found in Section 6.

##### 4.3.1. Assumptions on FLOP count of predominant operations

The FLOP count for the operations involved in the condensation and elementary approach are based upon the following assumptions:

- Arithmetic operations with individual elements have computational complexity  $\mathcal{O}(1)$ , as is the case with fixed-precision floating-point arithmetic. CPU time is equal for all individual operations (addition, multiplication etc.).
- Linear systems of equations are either solved using direct (factorization plus back-substitution) or iterative (preconditioning plus iterative solve) solution methods.
- The computational effort is dominated by the preprocessing and solve. Thus, all other operations are of negligible cost. In other words, we assume problems to be sufficiently large, that is  $n \gg 1$ .
- The properties (sparsity, symmetry, band density, definiteness, band structure etc.) of all system matrices are known. Based on this the solution method is chosen, as well as specific implementation(s).
- To calculate the gain  $\Xi_{\tilde{\beta}}$  the following are assumed to be known: problem size  $n$  and number of analysis sets  $a$  as well as the index sets  $\mathbb{F}^{(i)}$  and  $\mathbb{P}^{(i)}$ , indices of the DOFs of interest  $\mathbb{M}^{(i)}$ , prescribed state  $\mathbf{U}_p^{(i)}$  and applied load  $\mathbf{F}_f^{(i)}$  for each analysis set  $i \in \mathbb{A}$ .

The accuracy of  $\Xi_{\tilde{\beta}}$  with respect to  $\Xi_t$  under the given assumptions, as well as the range of validity will be shown by numerical experiments in Section 5.1.

#### 4.3.2. Theoretical approximated gain

An estimation of the FLOP count for solving a system of equations can, in general, be described by a function of the form

$$\tilde{\beta}[n, l] = \gamma[n] + \epsilon[n] \times l, \quad (51)$$

with  $n$  the system size,  $l$  the number of analyses, and  $\gamma[n]$  and  $\epsilon[n]$  system size dependent functions that are unique to the solution method. Here  $\gamma[n]$  is the FLOP count of the preprocessing, and  $\epsilon[n]$  the count per solve. Using such functions, we can approximate the expected gain of using the condensation approach compared to the elementary approach for a MPOP by

$$\Xi_{\tilde{\beta}}[n, m, a, l, b] \approx \frac{\sum_i^a \tilde{\beta}_s[n, l^{(i)} + b^{(i)}]}{\tilde{\beta}_s[n - m, m] + \sum_i^a \tilde{\beta}_d[m, l^{(i)} + b^{(i)}]}, \quad (52)$$

where the subscripts 's' and 'd' refer to the solution method specific functions for solving a sparse and dense system of equations, respectively. The parameter  $b^{(i)}$  denotes the number of non-zero right-hand-sides for the adjoint solve of analysis set  $i$ . The gain is based upon the response evaluation *and* sensitivity analysis, where we assume the preprocessing is reused where possible. In addition, we assumed that for a sufficiently large problems the ratio of prescribed DOFs to free DOFs becomes so small, that it is valid to assume the size of the analysis equals the system size, that is  $f^{(i)} \approx n$ ,  $\hat{f}^{(i)} \approx m$  for all  $i \in \mathbb{A}$ .

The ratio between the coefficients, that is  $\chi[n] := \frac{\gamma[n]}{\epsilon[n]}$ , is a predominant parameter that differentiates different solution methods, in the context of estimating the computational effort of problems for which the effort is dominated by linear solves. Direct solution methods are generally characterized with a high  $\chi$ . For extremely high  $\chi$ , such as direct computation of the inverse system matrix, the gain estimation becomes independent of the number of analyses. In that case, the gain is fully defined by the number and size of the analysis sets, that is Eq. (52) is approximated as

$$\Xi_{\tilde{\beta}}[n, m, a] \approx \frac{a \times \gamma_s[n]}{\gamma_s[n - m] + a \times \gamma_d[m]}, \quad (53)$$

with  $\gamma_s$  and  $\gamma_d$  the coefficients determining the preprocessing FLOP count for solving a sparse and dense system of equations, respectively.

Preconditioned iterative methods characteristically have a substantial lower  $\chi$ . For solution methods with an extremely low  $\chi$ , such as Jacobi preconditioned iterative methods, the gain estimation is dominated by the number of analyses irrespective of the number of analysis sets. That is, Eq. (52) may reasonably be approximated by

$$\Xi_{\tilde{\beta}}[n, m, l, b] \approx \frac{(l + b) \times \epsilon_s[n]}{m \times \epsilon_s[n - m] + (l + b) \times \epsilon_d[m]}, \quad (54)$$

with  $\epsilon_s$  and  $\epsilon_d$  the coefficients determining the FLOP count of solving a sparse and dense system of equations, respectively. For illustrational purpose, we assumed in both cases that the extreme values of  $\chi$  pertained to the solution methods of both sparse and dense solvers.

#### 4.3.3. Theoretical complexity of solution methods

The Cholesky factorization (CHOL) with subsequent back-substitutions is the most commonly applied direct solution method (both for sparse and dense matrices) to solve symmetric positive definite system matrices of small to moderate size. The FLOP count of a CHOL (without reordering) and subsequent back-substitutions of a symmetric sparse banded system matrix depends on dimensionality  $n$  and bandwidth  $k$ . Note that the bandwidth for discretized PDEs depends on the number of dimensions, e.g. in 2D  $k = \sqrt{n}$ , whereas in 3D  $k = n^{\frac{2}{3}}$  [47]. The dimensionality has a slight impact on the cost of the factorization; the factorization is relatively more expensive for 3D problems. Given the difference in number of factorizations between the condensation and elementary approach—at least for the solution method presented—higher estimated gains are expected for 3D problems. In this work we will—without loss of generality—study 2D examples, and hence use the function coefficients of CHOL as provided in Table 2.

One of the most common iterative solution methods to solve symmetric positive definite system of equations is to first construct an incomplete Cholesky factorization (ICHOL) preconditioner and then find the solution using a Conjugate Gradient (CG) algorithm [49]. The computational cost of the preconditioning (for simple preconditioners) is assumed negligible compared to the cost of the iterative solve. Therefore, in what follows, this contribution is

**Table 2**

Characteristic function coefficients of a variety of commonly applied solution methods [2–4,47,48]. The subscripts ‘d’ and ‘s’ refer to the dense and sparse implementations, respectively. Both iCHOL-CG and MG-CG have negligible preprocessing effort.

|                   | $\gamma [n]$     | $\epsilon [n]$     | $\chi [n]$            |
|-------------------|------------------|--------------------|-----------------------|
| CHOL <sub>d</sub> | $\frac{1}{3}n^3$ | $2n$               | $\frac{1}{6}n^2$      |
| CHOL <sub>s</sub> | $n^2$            | $2n^{\frac{3}{2}}$ | $\frac{1}{2}\sqrt{n}$ |
| iCHOL-PCG         |                  | $2n^2$ (2D)        |                       |
| MG-PCG            |                  | $\propto n$        |                       |

neglected. This assumption also emphasizes the usability of the method for a wide range of solution methods that meet this characteristic. Each CG iteration involves multiple operations, of which one matrix–vector product is dominating. For a  $k$ -banded matrix, the total FLOP count per iteration is thus  $2nk$  [3]. Thus, for the same system size  $n$ , 3D problems will have a higher FLOP count per iteration. The resulting additional computational effort is expected to have a slight impact on the computational gain for high ratios of  $\eta$ . The required number of CG iterations is approximately proportional to  $\sqrt{n}$ , see Table 2.

In modern large-scale topology optimization implementations Krylov subspace iterative solvers, in particular MultiGrid Preconditioned Conjugate Gradient (MG-PCG) methods, are commonly applied. Such solvers provide mesh-independent convergence as well as good parallel scalability [4]. Similar to iCHOL-CG methods, the preprocessing costs are relatively negligible. However, in contrast, the number of CG iterations is independent of the system size. As a result, MG-PCG can provide a solution of a linear system for a computational cost proportional to  $n$ , see Table 2. Assuming the same solution method to solve the dense problem, the MG-PCG will have similar gain for low to moderate  $\eta$  (as the cost of solving the dense system is negligible). For high  $\eta$  the advantageous scaling of MG-PCG (relative to iCHOL-CG) is expected to have a slight negative impact on the gain of MG-PCG compared to iCHOL-CG, as the increasing cost of solving the dense system will have a larger influence.

Note that in both functions  $\beta_s[n, l]$  and  $\beta_d[n, l]$  and coefficients therein depend on the type of solver, implementation and computer hardware, and are thus subjected to change. As such, the provided coefficients in Table 2 are not expected to be accurate; the values are intended to provide insight on the influence of the ratio  $\eta$  to the gain estimation for different solution methods.

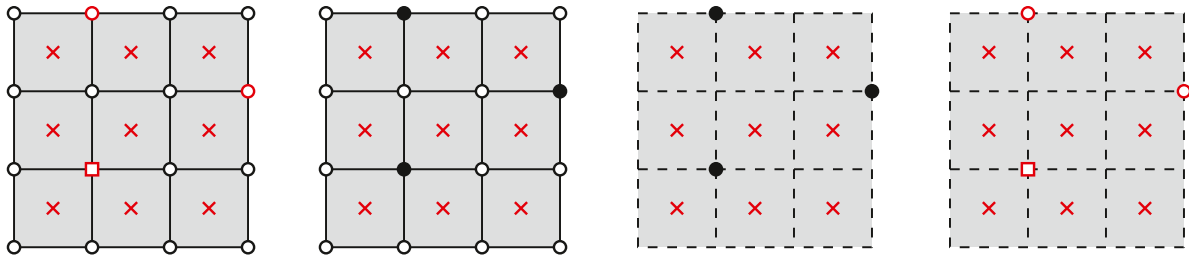
In order to show the general applicability, but keep the number of numerical examples to a minimum, the estimation is verified on numerical examples using two solution strategies with substantially different  $\chi$  to solve the system of equations. To represent problems of moderate size we first consider solving both sparse and dense system of equations using a direct solution method. To represent large-scale to very large-scale problems (in which memory storage and communication requirements are dominant), we employ an approach in which the sparse system of equations are solved using an iterative solution method. Considering the simplicity of implementation and ease of replication of results we use the iCHOL-CG. For the dense systems, the direct solution method remains the preferred choice.

## 5. Numerical examples

This section demonstrates the computational efficiency of the condensation approach for two different example problems, each representative for a specific type of MPOP.

We consider the following distinct examples:

1. A self-adjoint thermal conductivity MPOP where, by definition of the problem, the number of analysis sets equals the number of DOFs of interest, that is  $a = m$ , and each additional DOF of interest introduces an additional analysis set based on the reduced system matrix. This problem is representative for structural optimization problems with uncertainties in the use case or boundary conditions.
2. A non-self-adjoint displacement-based compliant mechanism design MPOP. The problem emphasizes the effect of reduction of computational effort of the sensitivity analysis. This example is representative for problems in which multiple response functions (requirements on behaviour from different use cases) all depend on a limited set of DOFs of interest.



**Fig. 6.** Schematic representation of Problem 1 for  $N = 9$  and  $m = 3$ . The DOFs of interest are chosen arbitrarily. From left to right: one of the three analyses as solved for in the elementary approach, subdivision into primary and secondary DOFs, resulting system after static condensation and, the equivalent analysis as solved for in the condensation approach. The DOFs with prescribed states are denoted by a square ( $\square$ ), the DOFs with free states by a circle ( $\circ$ ) and primary DOFs by a black dot ( $\bullet$ ) (analogously to Figs. 2 and 3). DOFs of interest are coloured in red ( $\circ$ ,  $\square$ ).

These examples are defined to easily allow variations of parameters to emphasize the validation of the efficiency improvements. Despite the conceptual description, the examples are representative for practical structural optimization problems.

Both examples are topology optimization problems, parametrized in a 2D structured grid of  $N$  bilinear quadrilaterals with a single design variable per element. The constrained nonlinear optimization problem is defined as

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && g_0[\mathbf{x}] \\ & \text{subject to} && g_j[\mathbf{x}] \leq 0, \quad j = 1, \dots, m \\ & && \mathbf{x} \in \mathbb{X}^N \end{aligned} \quad (55)$$

where  $\mathbf{x}$  is the field of design variables,  $\mathbb{X} := \{x_i \in \mathbb{R} \mid 0 < x_i \leq 1\}$ . The design variable field is blurred using a filter [50] with a radius of  $r = 2.0$  finite elements. The element conductivity/stiffness is related to the filtered design variable via the modified SIMP<sup>5</sup> interpolation function [51] with a constant penalty value of  $p = 3.0$ .

### 5.1. Problem 1

Consider the well known heat conduction topology optimization problem in a two-dimensional domain [52]. We investigate a variation of this problem with multiple analysis sets, by considering efficient heat conduction between  $m$  randomly distributed DOFs. In each of the  $a$  analysis sets, a single DOF acts as a heat sink (temperature prescribed to zero), and a heat load of random magnitude (between  $0 \text{ J s}^{-1}$  and  $1 \text{ J s}^{-1}$ ) is applied to the  $m - 1$  remaining primary DOFs.

Following the drawing conventions as previously used, a representation of this problem is sketched in Fig. 6. For this MPOP, the number of primary DOFs is equal to the number of distinct analysis sets, *i.e.*  $m = a$ . The aim is to find a design that efficiently conducts heat between all primary DOFs, for each considered scenario. Thereto, we minimize the overall conductive resistance for all considered analyses, that is

$$g_0[\mathbf{U}^{(1)}[\mathbf{x}], \dots, \mathbf{U}^{(a)}[\mathbf{x}]] = \sum_{i \in \mathbb{A}} \sum_j^{l^{(i)}} \mathbf{u}_j^{(i)} \cdot \mathbf{K}[\mathbf{x}] \mathbf{u}_j^{(i)} \quad (56)$$

for the elementary approach and

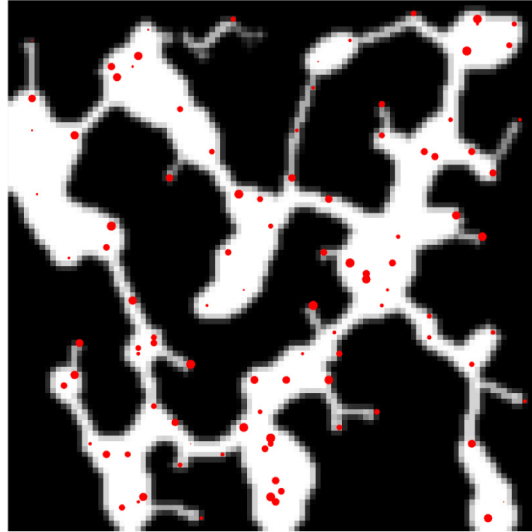
$$g_0[\hat{\mathbf{U}}^{(1)}[\mathbf{x}], \dots, \hat{\mathbf{U}}^{(a)}[\mathbf{x}]] = \sum_{i \in \mathbb{A}} \sum_j^{l^{(i)}} \hat{\mathbf{u}}_j^{(i)} \cdot \tilde{\mathbf{K}}[\mathbf{x}] \hat{\mathbf{u}}_j^{(i)} \quad (57)$$

for the condensation approach. The material usage is constrained as

$$g_1 = \sum_i^N \frac{\tilde{x}_i}{N\bar{v}} - 1, \quad (58)$$

<sup>5</sup> We omit further explanation as these terms, arguments and implications are considered common knowledge within the field.





**Fig. 7.** Resulting topology of Problem 1 as schematically sketched in Fig. 6. The white material is highly conductive, whereas the black material has low conductivity. For this MPOP the number of variables  $N = 100 \times 100$ , the number of structural DOFs  $n = 10000$ , and the number of primary DOFs and analyses  $m = a = 100$ . The maximum amount of material usage  $\bar{v} = 0.2$ . The location and magnitude (size of red dot) of the primary DOFs and applied loads are indicated in red.

where  $\bar{v}$  is the maximum ratio of solid to void material based upon the filtered field  $\tilde{\mathbf{x}}$ . An example of a resulting topology for  $m = 100$  is shown in Fig. 7, which can be reproduced using the attached MATLAB code.

#### 5.1.1. Computational efficiency

Although the choice of approach, *i.e.* elementary or condensation-based, has no influence on the resulting topology or design performance, it does influence the computational effort. For this specific MPOP  $l^{(i)} = m - 1$  for all  $i \in \mathbb{A}$  and  $\frac{d\mathbf{g}_0}{d\mathbf{U}^{(i)}} = \mathbf{F}^{(i)}$ . As a result, we can simply write  $\mathbf{A}^{(i)} = \mathbf{U}^{(i)}$  and fully equivalent for the condensation approach  $\hat{\mathbf{A}}^{(i)} = \hat{\mathbf{U}}^{(i)}$ . Hence, the problem is self-adjoint and there is no need to solve the additional system of equations of Eq. (48). Therefore,  $b^{(i)} = 0$  for all  $i \in \mathbb{A}$ . For the proposed optimization problem, Eq. (52) can thereto be simplified to

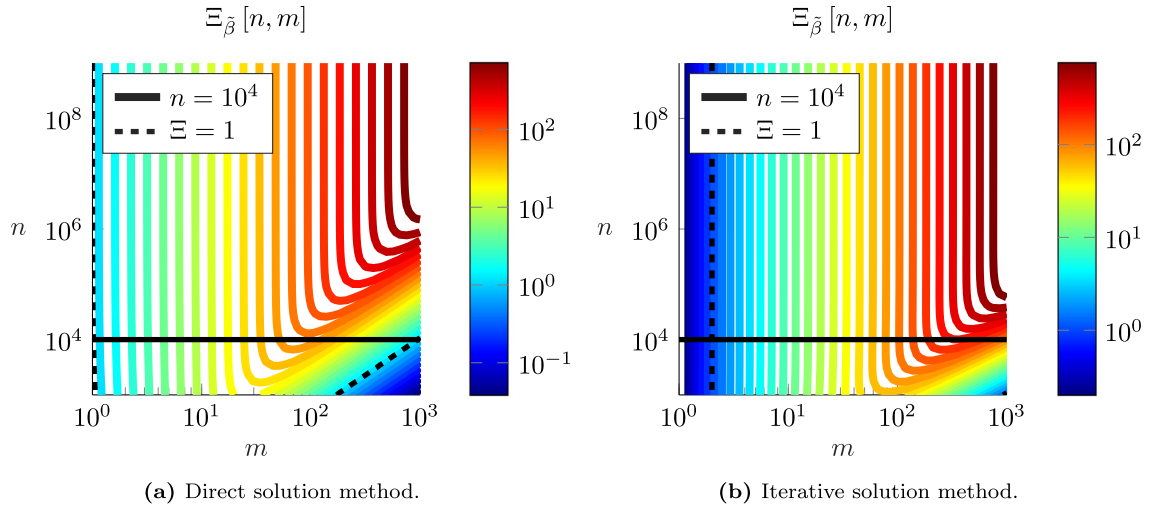
$$\Xi_{\tilde{\beta}}[n, m] \approx \frac{\sum_i^m \tilde{\beta}_s[n, m - 1]}{\tilde{\beta}_s[n - m, m] + \sum_i^m \tilde{\beta}_d[m, m - 1]}. \quad (59)$$

The computational gain  $\Xi_{\tilde{\beta}}$  as a function of  $n$  and  $m$  is plotted in Figs. 8(a) and 8(b) for direct and iterative solution methods respectively. The contour graphs show the condensation approach clearly outperforms the elementary approach, especially for large systems and small to moderate  $m$ . For this type of MPOP, a problem with many DOFs and a low ratio of  $m$  to  $n$  can be solved up to *1000 times faster* using the condensation approach independent of the chosen solution method. Figs. 9(a) and 9(b) give a clear view of the gain as a function of  $m$  for different system sizes using direct and iterative solution methods, respectively. In addition to the gain as a function of FLOP count  $\Xi_{\tilde{\beta}}$ , the gain obtained via a time measurement<sup>6</sup> for  $n = 10^4$  is plotted.<sup>7</sup> One may observe that  $\Xi_t$  slightly deviates from  $\Xi_{\tilde{\beta}}$ . Such deviation may reasonably be expected considering the severity and number of assumptions, as well as the uncertainties on the efficiency of the software, compilers and hardware used. As such, the analytical formulae, in its turn, are not expected to be an exact predictor of the results obtained by the time

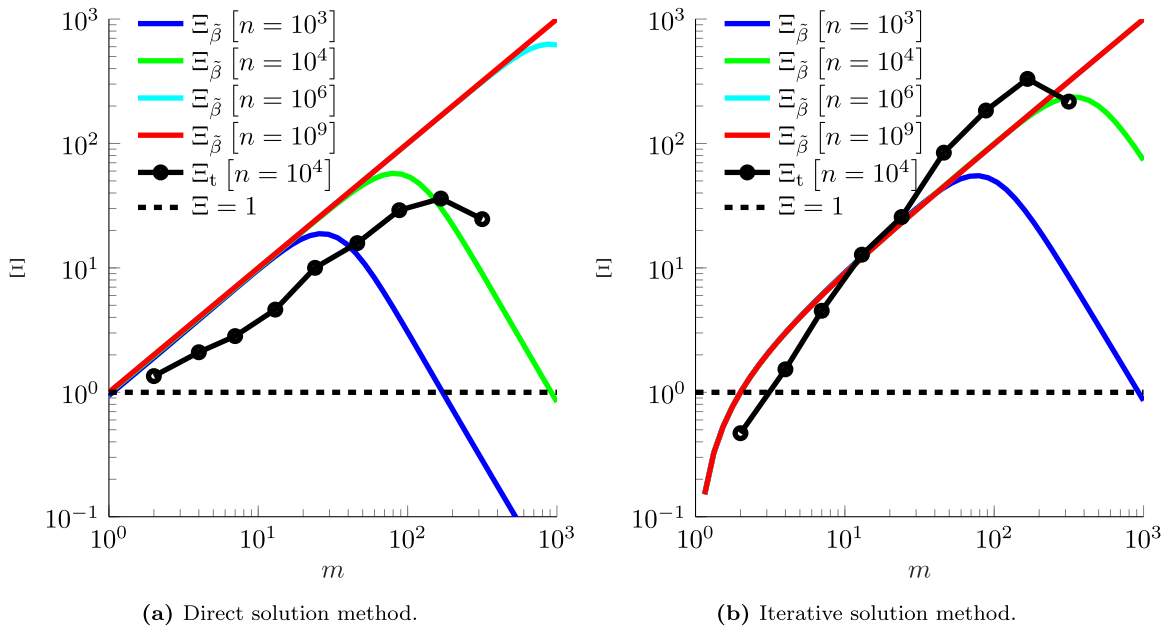
<sup>6</sup> The gain based on time measurements, equivalently to the gain based on FLOP count, only considers the time of solving the systems of equations.

<sup>7</sup> This problem size is considered sufficiently large to give a representative measure, without requiring unnecessary computation cost of the verification.



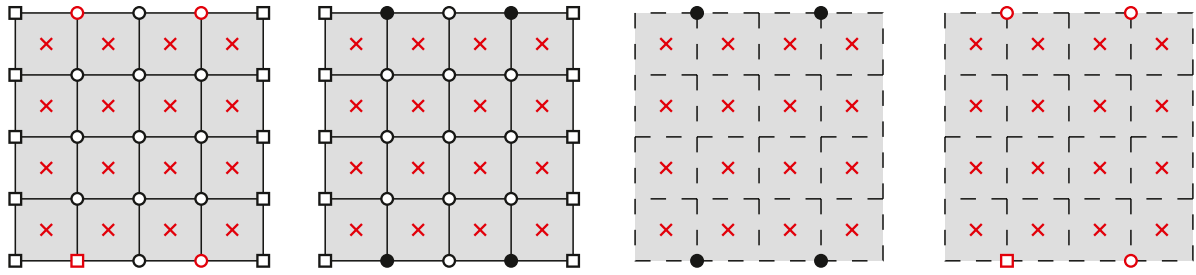


**Fig. 8.** Theoretical estimated gain of the condensation approach compared to the elementary approach for Problem 1. Figs. 8(a) and 8(b) show a triple-log contour plot of the gain  $\Xi_{\hat{\beta}}$  as a function of system size  $n$  and number of DOFs of interest  $m$  for the direct (a) and iterative (b) solution methods respectively.



**Fig. 9.** Theoretical estimated and experimental gain of the condensation approach compared to the elementary approach for Problem 1. Figs. 9(a) and 9(b) show double-log graphs of the gain as a function of the number of primary DOFs  $m$  for a variety of system sizes. The gain based on FLOP count is verified by calculating the gain based on CPU time as shown in Figs. 9(a) and 9(b).

measurements. Despite the deviations, the trend is captured. In line with the prognosis; there is an increasing gain up to some point, after which the gain decreases. This confirms that Eq. (52) can be used as a ‘order of magnitude’ estimator for the expected gain of using the condensation approach. In the following example, we will, thereto, utilize the analytical formulae without further verification.



**Fig. 10.** Schematic representation of Problem 2; a multi-input–multi-output compliant mechanism with  $N = 16$ , the number of inputs  $x = 2$  and  $m = 2x = 4$ . The number of DOFs per node is 2, however, for simplicity of visualization, only a single DOF is shown. From left to right (analogous to Fig. 6): one of the two analyses as solved for in the elementary approach, subdivision into primary and secondary DOFs, resulting system after static condensation and, the equivalent analysis as solved for in the condensation approach. Note that the displacements of both sides are fixed.

## 5.2. Problem 2

The second problem aims to maximize the amount of material required to achieve a displacement-based  $x$ -input  $x$ -output compliant mechanism satisfying prescribed input–output transmission ratios, see *e.g.* Larsen et al. [53]. The problem is sketched in Fig. 10. This second problem emphasizes the effect of reduction of computational effort of the sensitivity analysis.

The objective function for this MPOP is

$$g_0 = \sum_i^N \tilde{x}_i. \quad (60)$$

One can write the mechanism output displacements as a function of the input displacements as

$$\mathbf{u}_i = \mathbf{J}_{ij} \mathbf{u}_j, \quad \forall i, j = 1, \dots, x \quad (61)$$

where  $\mathbf{J}$  is the so-called Jacobian matrix,  $i$  denotes the output and  $j$  the input. For example,  $\mathbf{J}_{12}$  is the displacement of Output 1 due to an input displacement on Input 2. One requires  $a = x$  analyses with prescribed unit input displacements to obtain  $\mathbf{J}$ . In order to control all entries in the Jacobian matrix, we introduce  $2x$  transmission constraints, *i.e.*

$$g_{ij} = \frac{\mathbf{J}_{ij}}{\bar{\mathbf{J}}_{ij}} + 1, \quad (62)$$

with  $\bar{\mathbf{J}}_{ij} \neq 0$  the minimum transmission between input  $i$  and output  $j$ . Note this response can be used for both positive and negative ratios. If one assigns  $\bar{\mathbf{J}}_{ij} = u^*$ , then the required  $\mathbf{J}_{ij} \leq -u^*$  and if  $\bar{\mathbf{J}}_{ij} = -u^*$ , then  $\mathbf{J}_{ij} \geq u^*$ , for any target displacement  $u^* \in \mathbb{R}^+$ .

An example of the resulting design for  $x = 2$  and

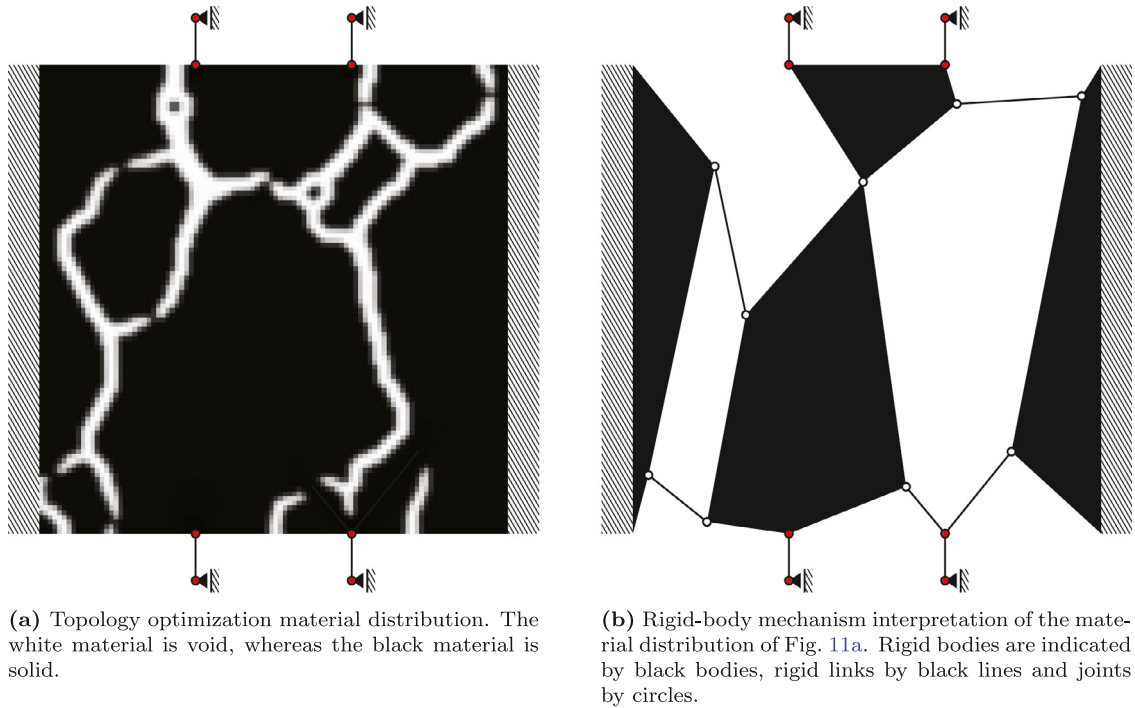
$$\bar{\mathbf{J}} = \begin{bmatrix} 0.5 & 2.0 \\ 1.0 & -1.0 \end{bmatrix}$$

is shown in Fig. 11. Due to the conflicting behaviour between volume maximization and minimum transmission ratio the resulting topology almost exactly satisfies the target Jacobian with a minimum amount of holes. To get an idea of the working principle, Fig. 11(b) provides a rigid-body mechanism interpretation of the material distribution of Fig. 11(a). One may vastly observe the desired behaviour is restricted to small displacements.

### 5.2.1. Computational efficiency

For this specific MPOP  $l^{(i)} = 1$  for all  $i \in \mathbb{A}$ . Since the problem is non-self-adjoint an additional  $x$  right-hand-sides per analysis has to be solved for, that is  $b^{(i)} = x$  for all  $i \in \mathbb{A}$ . Note that the number of DOFs of interest equals twice the number of inputs,<sup>8</sup> *i.e.*  $m = 2x$ . For the proposed optimization formulation, Eq. (52) can therefore

<sup>8</sup> Assuming the number of inputs and outputs are equal.



**Fig. 11.** Resulting topology and rigid-body mechanism interpretation of the MPOP posed in Section 5.2 as schematically sketched in Fig. 10. For this MPOP the number of variables  $N = 100 \times 100$ , the number of structural DOFs  $n = 10000$ , the number of analysis sets  $a = 2$  and the number of primary DOFs  $m = 2x = 4$ . The location of primary DOFs and applied loads are indicated in red. This multi-input–multi-output compliant mechanism is optimized for a Jacobian matrix  $\tilde{\mathbf{J}} = [0.5 \ 2.0; \ 1.0 \ -1.0]$ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

be simplified to

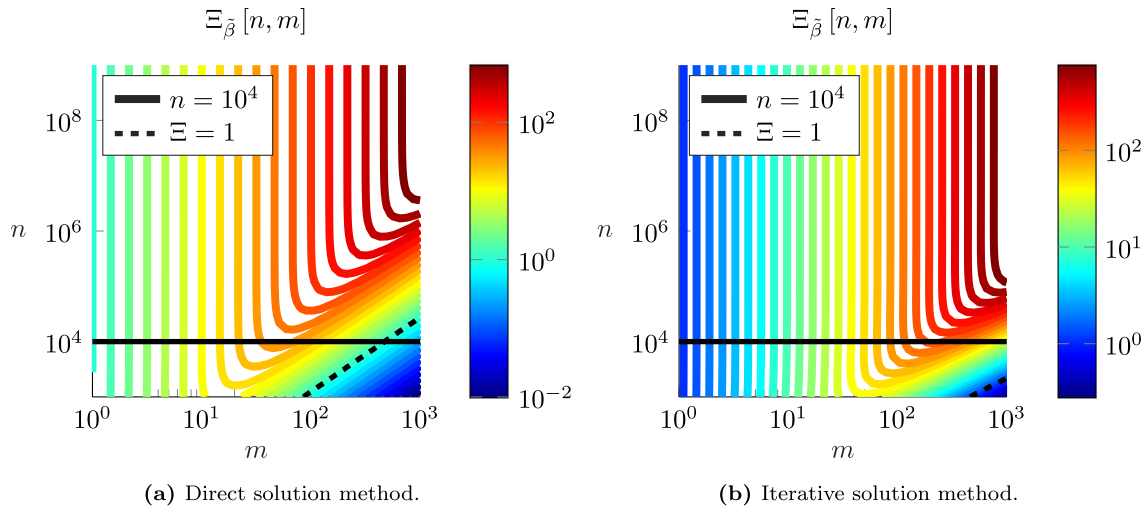
$$\Xi_{\tilde{\beta}}[n, m] \approx \frac{\sum_i^{\frac{m}{2}} \tilde{\beta}_s[n, m]}{\tilde{\beta}_s[n - m, m] + \sum_i^{\frac{m}{2}} \tilde{\beta}_d[m, m]}, \quad (63)$$

with  $m = 2x$  and  $x$  the number of inputs and outputs. Note the computational gain for this type of problem is twofold. One requires a single analysis set with  $2x$  analyses, as opposed to  $x$  analysis sets to obtain the responses plus an additional  $x$  analysis sets to obtain the sensitivities. Simple reasoning as well as the FLOP count analysis learns that for this type of problem the gain approximately equals the number of inputs (and thus the number of analysis sets) for low to moderate number of inputs, see Fig. 12.

## 6. Discussion

The proposed method is applicable to any linear(ized) PDE as well as multi-physical coupled problems and can be extended or adapted to a range of other problems. The method is still applicable to asymmetric system matrices. The advantageous property of inexpensive sensitivities vanishes for asymmetric matrices since the self-adjointness property no longer holds, hence the method is expected to reduce efficiency. The method is applicable to reduce the dimensionality of eigen-problems, with tremendous reduction in computational cost. However, in the reduced mass matrix, combinations of stiffness and mass elements appear. The result is that the eigenvalue–eigenvector problem is closely but not exactly preserved, with scenario-dependent accuracy issues [19].

A response function may depend on secondary DOFs, in which case an additional large system of linear equations has to be solved to obtain the sensitivities, see Appendix A.5. Although the additional effort is marginal since the preconditioner can be reused, it makes the use of responses dependent on secondary DOFs uneconomical. As opposed to the general guidelines presented in this work, one may want to put secondary DOFs with varying freedom



**Fig. 12.** Theoretical estimated gain of the condensation approach compared to the elementary approach for Problem 2. Figs. 12(a) and 12(b) show a triple-log contour plot of the gain  $\Xi_{\beta}$  as a function of system size  $n$  and number of DOFs of interest  $m$  for direct (a) and iterative (b) solution methods respectively.

per analysis set subjected to loading into the set of primary DOFs. This will increase the number of primary DOFs, however circumvents the necessity to calculate and use the reduced load. This will reduce the computational effort if the number of loaded secondary DOFs are small.

For scenarios with multiple DOFs following the same freedom *and* magnitude throughout each analysis set (for example the DOFs contained in a non-design domain), pre-processing of the system matrix by multi-point constraints<sup>9</sup> is recommended to limit the dimensionality of the reduced system matrix. This has no further consequences apart from implementation, possibly reduction in global dimensionality and change of sparsity pattern/band density.

For large-scale 3D problems, the memory requirements and parallelization generally become critical aspects. Using the elementary method requires storage of multiple (equal to the number of analysis sets  $a$ ) large system matrices (either factorizations or preconditioners), whereas the condensation approach only has to store a single system matrix of substantial dimension. Just as the elementary method, the condensation method consists of linear solves, matrix–vector products and dot-products. Thereto, no substantial differences in terms of ‘ease of parallelization’ are expected. For very large-scale problems it is recommended to distribute the condensation solves, Eqs. (20) and (21), over multiple computing nodes (as opposed to distribute each solve over multiple nodes) to reduce the amount of communication.

## 7. Conclusion

This study aims to find a solution to efficiently solve for large-scale structural optimization problems involving multiple analyses with varying boundary conditions. With an increasing number of analyses, the computational effort to solve such problems quickly increases. Using static condensation to eliminate the DOFs that do not change freedom between analyses *and* are not of interest to any of the optimization responses, the effort can be highly reduced. In contrast to solving for multiple systems of equations with different sparse system matrices of high dimensionality, this allows for solving multiple systems of equations with *the same* sparse system matrix of high dimensionality *and* multiple systems of equations with different dense system matrices of low dimensionality. A profound finding is that the sensitivities of the static condensation procedure can be calculated without solving additional systems of equations, even if the response functions are not self-adjoint. Consequently, the dimensionality of the adjoint equations is highly reduced.

<sup>9</sup> Primary–secondary, penalty augmentation or Lagrange multiplier adjunction methods.

Numerical examples indicate that an algorithmic complexity analysis can be used to roughly estimate the expected computational effort of a broad range of problems, even if solely the major contributors are taken into account. In line with the prognosis the numerical examples demonstrate that substantial computational gains can be achieved. However, the presented examples only cover a small part of the total applicability.

It is expected that future research will reveal more applications that benefit from this method. We expect at least applications in structural and multidisciplinary optimization with many conflicting requirements based on a variety of load cases. Given the limited additional implementation effort, future work on—or including—multi-partition structural optimization problem(s) should consider a back-of-the-envelope estimation of computational gain by using the condensation approach.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

The authors would like to thank Freek Broeren (set theory), Can Ayas (Einstein notation), Arnoud Delissen (dyadic product) and Max van der Kolk (algorithmic complexity analysis) for their help. This work was supported by the Dutch Research Council (NWO), The Netherlands Applied and Engineering Sciences (AES), The Netherlands project 16191 entitled Stable and Adjustable Mechanisms for Optical Instruments and Implants (SAMOI).

### Appendix A. Sensitivity analysis

Dependencies will, after first mention, be omitted for brevity without further notice.

#### A.1. Sensitivities of the reduced system matrix

Consider a response function  $g[\tilde{\mathbf{K}}[\mathbf{K}]]$ , with

$$\tilde{\mathbf{K}} := \hat{\mathbf{K}} - \check{\mathbf{K}}_f \check{\mathbf{X}}_f. \quad (18 \text{ revisited})$$

Here the system matrix is a function of the design variables, *i.e.*  $\mathbf{K}[\mathbf{x}]$ . For now we are solely interested in the sensitivities of the reduced system matrix with respect to the system matrix and therefore omit this dependency from here onward. The full derivative can be written as

$$dg[\tilde{\mathbf{K}}[\mathbf{K}]] = \frac{dg}{d\tilde{\mathbf{K}}} : \frac{d\tilde{\mathbf{K}}}{d\mathbf{K}} : d\mathbf{K}, \quad (31 \text{ revisited})$$

where  $\frac{dg}{d\tilde{\mathbf{K}}}$  is the sensitivity of the response function with respect to  $\tilde{\mathbf{K}}$  and  $d\mathbf{K}$  the sensitivity of the system matrix.

In the following we assume both are known. The aim is to find the fourth order tensor  $\frac{d\tilde{\mathbf{K}}}{d\mathbf{K}}$ , *i.e.* the sensitivities of the reduced system matrix with respect to the system matrix.

##### A.1.1. Direct method

Direct differentiation of Eq. (18) yields

$$d\tilde{\mathbf{K}} = d\hat{\mathbf{K}} - d\check{\mathbf{K}}_f \check{\mathbf{X}}_f - \check{\mathbf{K}}_f d\check{\mathbf{X}}_f. \quad (32 \text{ revisited})$$

Here  $d\check{\mathbf{X}}_f$  can be obtained by direct differentiation of Eq. (20), that is

$$d\check{\mathbf{X}}_f = -\check{\mathbf{K}}_{ff}^{-1} d\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f + \check{\mathbf{K}}_{ff}^{-1} d\check{\mathbf{K}}_f. \quad (33 \text{ revisited})$$

Substitution of Eq. (33) into Eq. (32) gives

$$d\tilde{\mathbf{K}} = d\hat{\mathbf{K}} - d\check{\mathbf{K}}_f \check{\mathbf{X}}_f + \check{\mathbf{X}}_f d\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f - \check{\mathbf{X}}_f d\check{\mathbf{K}}_f. \quad (A.1)$$

Substitution of Eq. (30) into Eq. (A.1) yields

$$d\tilde{\mathbf{K}} = \hat{\mathbf{S}}^T d\mathbf{K}\hat{\mathbf{S}} - \hat{\mathbf{S}}^T d\mathbf{K}\hat{\mathbf{S}}_f^{\vee} \hat{\mathbf{X}}_f + \hat{\mathbf{X}}_f^{\vee T} d\mathbf{K}\hat{\mathbf{S}}_f^{\vee} \hat{\mathbf{X}}_f - \hat{\mathbf{X}}_f^{\vee T} d\mathbf{K}\hat{\mathbf{S}}. \quad (\text{A.2})$$

Subsequent substitution of the result into Eq. (31) gives the expression for the sensitivities of the response function, *i.e.*

$$\begin{aligned} dg &= \frac{dg}{d\tilde{\mathbf{K}}} : \left( \left( \hat{\mathbf{S}}^T - \hat{\mathbf{X}}_f^{\vee T} \right) d\mathbf{K} \left( \hat{\mathbf{S}} - \hat{\mathbf{S}}_f^{\vee} \hat{\mathbf{X}}_f \right) \right) \\ &= \frac{dg}{d\tilde{\mathbf{K}}} : \left( \mathbf{A}^T d\mathbf{K} \mathbf{A} \right), \end{aligned} \quad (34 \text{ revisited})$$

where

$$\mathbf{A} := \hat{\mathbf{S}} - \hat{\mathbf{S}}_f^{\vee} \hat{\mathbf{X}}_f. \quad (36 \text{ revisited})$$

We can simplify<sup>10</sup> this to obtain

$$dg = \left( \mathbf{A} \frac{dg}{d\tilde{\mathbf{K}}} \otimes \mathbf{A} \right) : d\mathbf{K}, \quad (\text{A.3})$$

In addition, rewriting<sup>3</sup> of Eq. (34) gives

$$\frac{d\tilde{\mathbf{K}}}{d\mathbf{K}} = \mathbf{A} \otimes \mathbf{A}. \quad (37 \text{ revisited})$$

#### A.1.2. Adjoint method

Alternatively to the direct method, one can use the adjoint method [43,45]. After augmentation of the response function as defined in Eq. (31) with Eq. (20), we get

$$\mathcal{L}[\tilde{\mathbf{K}}[\mathbf{K}], \hat{\mathbf{A}}_f] := g[\tilde{\mathbf{K}}[\mathbf{K}]] + \hat{\mathbf{A}}_f : \left( \hat{\mathbf{K}}_{ff}^{\vee} \hat{\mathbf{X}}_f - \hat{\mathbf{K}}_f^{\vee} \right), \quad (38 \text{ revisited})$$

where  $\mathcal{L}[\tilde{\mathbf{K}}[\mathbf{K}], \hat{\mathbf{A}}_f]$  is the Lagrangian and  $\hat{\mathbf{A}}_f \in \mathbb{R}^{\check{f} \times m}$  the Lagrange multipliers related to  $\hat{\mathbf{X}}_f$ . Full differentiation of both sides gives

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{K}}} : d\tilde{\mathbf{K}} + \hat{\mathbf{A}}_f : \left( d\hat{\mathbf{K}}_{ff}^{\vee} \hat{\mathbf{X}}_f + \hat{\mathbf{K}}_{ff}^{\vee} d\hat{\mathbf{X}}_f - d\hat{\mathbf{K}}_f^{\vee} \right). \quad (\text{A.4})$$

Substitution of Eq. (32) into Eq. (A.4) yields

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{K}}} : \left( d\hat{\mathbf{K}} - d\hat{\mathbf{K}}_f^{\vee} \hat{\mathbf{X}}_f - \hat{\mathbf{K}}_f^{\vee} d\hat{\mathbf{X}}_f \right) + \hat{\mathbf{A}}_f : \left( d\hat{\mathbf{K}}_{ff}^{\vee} \hat{\mathbf{X}}_f + \hat{\mathbf{K}}_{ff}^{\vee} d\hat{\mathbf{X}}_f - d\hat{\mathbf{K}}_f^{\vee} \right). \quad (\text{A.5})$$

Since  $d\hat{\mathbf{X}}_f$  is computationally expensive to calculate, the aim is to circumvent this by a suitable choice of  $\hat{\mathbf{A}}_f$ . Collecting terms involving  $d\hat{\mathbf{X}}_f$  gives

$$\hat{\mathbf{A}}_f : \left( \hat{\mathbf{K}}_{ff}^{\vee} d\hat{\mathbf{X}}_f \right) - \frac{dg}{d\tilde{\mathbf{K}}} : \left( \hat{\mathbf{K}}_f^{\vee} d\hat{\mathbf{X}}_f \right). \quad (\text{A.6})$$

This can be rewritten<sup>11</sup> into

$$\left( \hat{\mathbf{K}}_{ff}^{\vee} \hat{\mathbf{A}}_f - \hat{\mathbf{K}}_f^{\vee} \frac{dg}{d\tilde{\mathbf{K}}} \right) : d\hat{\mathbf{X}}_f. \quad (\text{A.7})$$

To make the terms involving  $d\hat{\mathbf{X}}_f$  vanish, the term in parentheses must be zero. This can be achieved by choosing  $\hat{\mathbf{A}}_f$  as

$$\hat{\mathbf{A}}_f = \hat{\mathbf{K}}_{ff}^{\vee -1} \hat{\mathbf{K}}_f^{\vee} \frac{dg}{d\tilde{\mathbf{K}}} = \hat{\mathbf{X}}_f \frac{dg}{d\tilde{\mathbf{K}}}. \quad (41 \text{ revisited})$$

<sup>10</sup>  $\mathbf{A} : (\mathbf{BCD}) = (\mathbf{B}^T \mathbf{A} \otimes \mathbf{D}) : \mathbf{C}$ .

<sup>11</sup>  $\mathbf{A} : (\mathbf{BC}) = (\mathbf{B}^T \mathbf{A}) : \mathbf{C}$ .

Here the self-adjointness property appears; no additional solves are needed to compute  $\check{\check{\Lambda}}_f$ . With these multipliers, the terms involving  $d\check{\check{\mathbf{X}}}_f$  vanish from Eq. (A.5), that is

$$d\mathcal{L} = \frac{dg}{d\check{\check{\mathbf{K}}}} : \left( d\check{\check{\mathbf{K}}} - d\check{\check{\mathbf{K}}}_f \check{\check{\mathbf{X}}}_f \right) + \check{\check{\Lambda}}_f : \left( d\check{\check{\mathbf{K}}}_{ff} \check{\check{\mathbf{X}}}_f - d\check{\check{\mathbf{K}}}_f \right), \quad (\text{A.8})$$

with  $\check{\check{\Lambda}}_f$  as defined in Eq. (41). Next, substitution of Eq. (30) into Eq. (A.8) yields

$$\begin{aligned} d\mathcal{L} &= \frac{dg}{d\check{\check{\mathbf{K}}}} : \left( \hat{\mathbf{S}}^T d\mathbf{K}\hat{\mathbf{S}} - \hat{\mathbf{S}}^T d\mathbf{K}\check{\check{\mathbf{S}}}_f \check{\check{\mathbf{X}}}_f \right) + \check{\check{\Lambda}}_f : \left( \check{\check{\mathbf{S}}}_f^T d\mathbf{K}\check{\check{\mathbf{S}}}_f \check{\check{\mathbf{X}}}_f - \check{\check{\mathbf{S}}}_f^T d\mathbf{K}\hat{\mathbf{S}} \right) \\ &= \frac{dg}{d\check{\check{\mathbf{K}}}} : \left( \hat{\mathbf{S}}^T d\mathbf{K}\mathbf{A} \right) - \check{\check{\Lambda}}_f : \left( \check{\check{\mathbf{S}}}_f^T d\mathbf{K}\mathbf{A} \right) \\ &= \left( \hat{\mathbf{S}} \frac{dg}{d\check{\check{\mathbf{K}}}} - \check{\check{\mathbf{S}}}_f^T \check{\check{\Lambda}}_f \right)^T d\mathbf{K}\mathbf{A}. \end{aligned} \quad (\text{39 revised})$$

The reader can verify that substitution of Eq. (41) into Eq. (39) results in Eq. (A.3).

## A.2. Sensitivities of the reduced load

Consider a response function  $g \left[ \tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\check{\mathbf{F}}}_f, \check{\check{\mathbf{U}}}_p \right] \right]$ , with

$$\tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\check{\mathbf{F}}}_f, \check{\check{\mathbf{U}}}_p \right] := \check{\check{\mathbf{K}}}_f \check{\check{\mathbf{V}}}_f \left[ \mathbf{K}, \check{\check{\mathbf{F}}}_f, \check{\check{\mathbf{U}}}_p \right] - \check{\check{\mathbf{K}}}_p \check{\check{\mathbf{U}}}_p. \quad (\text{19 revisited})$$

By applying the chain-rule, the full derivative of this response function can be written as

$$\begin{aligned} dg \left[ \tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\check{\mathbf{F}}}_f, \check{\check{\mathbf{U}}}_p \right] \right] &= \frac{dg}{d\tilde{\mathbf{F}}} : d\tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\check{\mathbf{F}}}_f, \check{\check{\mathbf{U}}}_p \right] \\ &= \frac{dg}{d\tilde{\mathbf{F}}} : \left( \frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{K}} : d\mathbf{K} + \frac{\partial \tilde{\mathbf{F}}}{\partial \check{\check{\mathbf{F}}}_f} : d\check{\check{\mathbf{F}}}_f + \frac{\partial \tilde{\mathbf{F}}}{\partial \check{\check{\mathbf{U}}}_p} : d\check{\check{\mathbf{U}}}_p \right). \end{aligned} \quad (\text{A.9})$$

The aim is to find  $\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{K}}$ ,  $\frac{\partial \tilde{\mathbf{F}}}{\partial \check{\check{\mathbf{F}}}_f}$  and  $\frac{\partial \tilde{\mathbf{F}}}{\partial \check{\check{\mathbf{U}}}_p}$ .

### A.2.1. Direct method

Direct differentiation of both sides of Eq. (19) yields

$$d\tilde{\mathbf{F}} = d\check{\check{\mathbf{K}}}_f \check{\check{\mathbf{V}}}_f + \check{\check{\mathbf{K}}}_f d\check{\check{\mathbf{V}}}_f - d\check{\check{\mathbf{K}}}_p \check{\check{\mathbf{U}}}_p - \check{\check{\mathbf{K}}}_p d\check{\check{\mathbf{U}}}_p. \quad (\text{A.10})$$

Here  $d\check{\check{\mathbf{V}}}_f$  can be obtained by direct differentiation of Eq. (21), that is

$$d\check{\check{\mathbf{V}}}_f = \check{\check{\mathbf{K}}}_{ff}^{-1} \left( -d\check{\check{\mathbf{K}}}_{ff} \check{\check{\mathbf{V}}}_f + d\check{\check{\mathbf{K}}}_{fp} \check{\check{\mathbf{U}}}_p - d\check{\check{\mathbf{F}}}_f + \check{\check{\mathbf{K}}}_{fp} d\check{\check{\mathbf{U}}}_p \right). \quad (\text{A.11})$$

Substitution of Eq. (A.11) into Eq. (A.10), and using the definition of  $\check{\check{\mathbf{V}}}_f$  from Eq. (21) yields

$$d\tilde{\mathbf{F}} = d\check{\check{\mathbf{K}}}_f \check{\check{\mathbf{V}}}_f + \check{\check{\mathbf{X}}}_f \left( -d\check{\check{\mathbf{K}}}_{ff} \check{\check{\mathbf{V}}}_f + d\check{\check{\mathbf{K}}}_{fp} \check{\check{\mathbf{U}}}_p - d\check{\check{\mathbf{F}}}_f + \check{\check{\mathbf{K}}}_{fp} d\check{\check{\mathbf{U}}}_p \right) - d\check{\check{\mathbf{K}}}_p \check{\check{\mathbf{U}}}_p - \check{\check{\mathbf{K}}}_p d\check{\check{\mathbf{U}}}_p. \quad (\text{A.12})$$

For conciseness, we define the variable

$$\mathbf{C}^T := \check{\check{\mathbf{X}}}_f \check{\check{\mathbf{K}}}_{fp} - \check{\check{\mathbf{K}}}_p, \quad (\text{A.13})$$

and substitute this into Eq. (A.12), which yields

$$d\tilde{\mathbf{F}} = d\check{\check{\mathbf{K}}}_f \check{\check{\mathbf{V}}}_f - \check{\check{\mathbf{X}}}_f d\check{\check{\mathbf{K}}}_{ff} \check{\check{\mathbf{V}}}_f + \check{\check{\mathbf{X}}}_f d\check{\check{\mathbf{K}}}_{fp} \check{\check{\mathbf{U}}}_p - d\check{\check{\mathbf{K}}}_p \check{\check{\mathbf{U}}}_p - \check{\check{\mathbf{X}}}_f d\check{\check{\mathbf{F}}}_f + \mathbf{C}^T d\check{\check{\mathbf{U}}}_p. \quad (\text{A.14})$$

Subsequent substitution of Eq. (30) into Eq. (A.14), and Eq. (A.14) into Eq. (A.9) gives an expression for the sensitivities of the response function, *i.e.*

$$dg = \frac{dg}{d\tilde{\mathbf{F}}} : \left( \hat{\mathbf{S}}^T d\mathbf{K}\check{\check{\mathbf{S}}}_f \check{\check{\mathbf{V}}}_f - \check{\check{\mathbf{X}}}_f \hat{\mathbf{S}}_f^T d\mathbf{K}\check{\check{\mathbf{S}}}_f \check{\check{\mathbf{V}}}_f + \check{\check{\mathbf{X}}}_f \hat{\mathbf{S}}_f^T d\mathbf{K}\check{\check{\mathbf{S}}}_p \check{\check{\mathbf{U}}}_p - \hat{\mathbf{S}}^T d\mathbf{K}\check{\check{\mathbf{S}}}_p \check{\check{\mathbf{U}}}_p - \check{\check{\mathbf{X}}}_f d\check{\check{\mathbf{F}}}_f + \mathbf{C}^T d\check{\check{\mathbf{U}}}_p \right). \quad (\text{A.15})$$



We can simplify<sup>10</sup> this to obtain

$$dg = \left( \mathbf{A} \frac{dg}{d\tilde{\mathbf{F}}} \otimes \mathbf{B} \right) : d\mathbf{K} - \check{\mathbf{X}}_f \frac{dg}{d\tilde{\mathbf{F}}} : d\check{\mathbf{F}}_f + \mathbf{C} \frac{dg}{d\tilde{\mathbf{F}}} : d\check{\mathbf{U}}_p, \quad (\text{A.16})$$

with

$$\mathbf{B} := \check{\mathbf{S}}_f \check{\mathbf{V}}_f - \check{\mathbf{S}}_p \check{\mathbf{U}}_p. \quad (\text{43 revised})$$

And thus

$$\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{K}} = \mathbf{A} \otimes \mathbf{B}. \quad (\text{42 revised})$$

Note that all terms in Eq. (A.16) are calculated in the corresponding forward analysis, and thus no additional solves are required.

### A.2.2. Adjoint method

After augmentation of the response function with the linear system of equations from Eq. (21), we get

$$\mathcal{L} \left[ \tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p \right], \check{\mathbf{A}}_f \right] := g \left[ \tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p \right] \right] - \check{\mathbf{A}}_f : \left( \check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f + \check{\mathbf{F}}_f - \check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p \right), \quad (\text{A.17})$$

where  $\check{\mathbf{A}}_f \in \mathbb{R}^{\check{J} \times I}$  are the Lagrange multipliers related to  $\check{\mathbf{V}}_f$ . Differentiation of both sides gives

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{F}}} : d\tilde{\mathbf{F}} - \check{\mathbf{A}}_f : (d\check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f + \check{\mathbf{K}}_{ff} d\check{\mathbf{V}}_f - d\check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p + d\check{\mathbf{F}}_f - \check{\mathbf{K}}_{fp} d\check{\mathbf{U}}_p). \quad (\text{A.18})$$

Substitution of Eq. (A.10) into Eq. (A.18) gives

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{F}}} : (d\check{\mathbf{K}}_f \check{\mathbf{V}}_f + \check{\mathbf{K}}_f d\check{\mathbf{V}}_f - d\check{\mathbf{K}}_p \check{\mathbf{U}}_p - \check{\mathbf{K}}_p d\check{\mathbf{U}}_p) - \check{\mathbf{A}}_f : (d\check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f + \check{\mathbf{K}}_{ff} d\check{\mathbf{V}}_f - d\check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p + d\check{\mathbf{F}}_f - \check{\mathbf{K}}_{fp} d\check{\mathbf{U}}_p). \quad (\text{A.19})$$

Since  $d\check{\mathbf{V}}_f$  is computationally expensive to calculate, aim is to circumvent this by a suitable choice of  $\check{\mathbf{A}}_f$ . Collecting terms involving  $d\check{\mathbf{V}}_f$  gives

$$\frac{dg}{d\tilde{\mathbf{F}}} : (\check{\mathbf{K}}_f d\check{\mathbf{V}}_f) - \check{\mathbf{A}}_f : (\check{\mathbf{K}}_{ff} d\check{\mathbf{V}}_f). \quad (\text{A.20})$$

This can be rewritten<sup>11</sup> into

$$\left( \check{\mathbf{K}} \frac{dg}{d\tilde{\mathbf{F}}} - \check{\mathbf{K}} \check{\mathbf{A}}_f \right) : d\check{\mathbf{V}}_f. \quad (\text{A.21})$$

To make the terms  $d\check{\mathbf{V}}_f$  vanish we choose  $\check{\mathbf{A}}_f$  as

$$\check{\mathbf{A}}_f = \check{\mathbf{K}}_{ff}^{-1} \check{\mathbf{K}}_f \frac{dg}{d\tilde{\mathbf{F}}} = \check{\mathbf{X}}_f \frac{dg}{d\tilde{\mathbf{F}}}. \quad (\text{A.22})$$

Note again that no adjoint solve is necessary. Substitution of Eq. (30) into Eq. (A.19) yields

$$d\mathcal{L} = \frac{dg}{d\tilde{\mathbf{F}}} : (\hat{\mathbf{S}}^T d\mathbf{K} \check{\mathbf{S}}_f \check{\mathbf{V}}_f - \hat{\mathbf{S}}^T d\mathbf{K} \check{\mathbf{S}}_p \check{\mathbf{U}}_p - \hat{\mathbf{K}}_p d\check{\mathbf{U}}_p) - \check{\mathbf{A}}_f : (\check{\mathbf{S}}_f^T d\mathbf{K} \check{\mathbf{S}}_f \check{\mathbf{V}}_f - \check{\mathbf{S}}_f^T d\mathbf{K} \check{\mathbf{S}}_p \check{\mathbf{U}}_p + d\check{\mathbf{F}}_f - \check{\mathbf{K}}_{fp} d\check{\mathbf{U}}_p), \quad (\text{A.23})$$

with  $\check{\mathbf{A}}_f$  as defined in Eq. (A.22). The reader can verify that substitution of Eq. (A.22) into Eq. (A.23) results in Eq. (A.16).

### A.3. Sensitivities of the state

Often responses are a function of the state, *i.e.*  $g \left[ \hat{\mathbf{U}}_f \left[ \check{\mathbf{K}} \left[ \mathbf{K} \right], \tilde{\mathbf{F}} \left[ \mathbf{K}, \check{\mathbf{F}}_f, \check{\mathbf{U}}_p \right], \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p \right] \right]$ . Note that this may be the state of multiple analyses, that is  $g \left[ \hat{\mathbf{U}}_f^{(1)}, \hat{\mathbf{U}}_f^{(2)}, \dots, \hat{\mathbf{U}}_f^{(a)} \right]$ , where  $a$  is the total number of analyses based upon  $\check{\mathbf{K}}$ .

Therefore, one is required to determine  $d\hat{\mathbf{U}}_f^{(i)} [\tilde{\mathbf{K}}, \tilde{\mathbf{F}}, \hat{\mathbf{F}}_f^{(i)}, \hat{\mathbf{U}}_p^{(i)}]$  for all  $i \in \mathbb{A}$ . For clarity, we omit the superscript  $i$  from here on (assume a single analysis set). Taking the full derivative of  $g[\hat{\mathbf{U}}_f, \tilde{\mathbf{K}}, \tilde{\mathbf{F}}, \hat{\mathbf{F}}_f, \hat{\mathbf{U}}_p]$  yields

$$dg = \frac{dg}{d\hat{\mathbf{U}}_f} : d\hat{\mathbf{U}}_f = \frac{dg}{d\hat{\mathbf{U}}_f} : \left( \frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{K}}} : d\tilde{\mathbf{K}} + \frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{F}}} : d\tilde{\mathbf{F}} + \frac{\partial \hat{\mathbf{U}}_f}{\partial \hat{\mathbf{F}}_f} : d\hat{\mathbf{F}}_f + \frac{\partial \hat{\mathbf{U}}_f}{\partial \hat{\mathbf{U}}_p} : d\hat{\mathbf{U}}_p \right), \quad (44 \text{ revisited})$$

where we assume  $\frac{dg}{d\hat{\mathbf{U}}_f}$  to be known. The terms  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{K}}}$ ,  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \tilde{\mathbf{F}}}$ ,  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \hat{\mathbf{F}}_f}$  and  $\frac{\partial \hat{\mathbf{U}}_f}{\partial \hat{\mathbf{U}}_p}$  are to be determined. To this end, we augment the response function with the linear system of equations from Eq. (27), which yields

$$\mathcal{L}[\hat{\mathbf{U}}_f, \hat{\mathbf{A}}_f] := g[\hat{\mathbf{U}}_f] - \hat{\mathbf{A}}_f : (\tilde{\mathbf{K}}_{ff}\hat{\mathbf{U}}_f - \hat{\mathbf{F}}_f + \tilde{\mathbf{K}}_{fp}\hat{\mathbf{U}}_p - \tilde{\mathbf{F}}_f) \quad (A.24)$$

where  $\mathcal{L}[\hat{\mathbf{U}}_f, \hat{\mathbf{A}}_f]$  is the Lagrangian and  $\hat{\mathbf{A}}_f \in \mathbb{R}^{\hat{f} \times l}$  the Lagrange multipliers related to  $\hat{\mathbf{U}}_f$ . Full differentiation of both sides gives

$$d\mathcal{L} = \frac{dg}{d\hat{\mathbf{U}}_f} : d\hat{\mathbf{U}}_f - \hat{\mathbf{A}}_f : (d\tilde{\mathbf{K}}_{ff}\hat{\mathbf{U}}_f + \tilde{\mathbf{K}}_{ff}d\hat{\mathbf{U}}_f - d\hat{\mathbf{F}}_f + d\tilde{\mathbf{K}}_{fp}\hat{\mathbf{U}}_p + \tilde{\mathbf{K}}_{fp}d\hat{\mathbf{U}}_p - d\tilde{\mathbf{F}}_f). \quad (A.25)$$

Since  $d\hat{\mathbf{U}}_f$  is computationally expensive to calculate, aim is to circumvent this by a suitable choice of  $\hat{\mathbf{A}}_f$ . Collecting terms involving  $d\hat{\mathbf{U}}_f$  gives

$$\frac{dg}{d\hat{\mathbf{U}}_f} : d\hat{\mathbf{U}}_f - \hat{\mathbf{A}}_f : \tilde{\mathbf{K}}_{ff}d\hat{\mathbf{U}}_f. \quad (A.26)$$

This can be rewritten<sup>11</sup> into

$$\left( \frac{dg}{d\hat{\mathbf{U}}_f} - \tilde{\mathbf{K}}_{ff}\hat{\mathbf{A}}_f \right) : d\hat{\mathbf{U}}_f. \quad (A.27)$$

To make the terms  $d\hat{\mathbf{U}}_f$  vanish we choose  $\hat{\mathbf{A}}_f$  as

$$\tilde{\mathbf{K}}_{ff}\hat{\mathbf{A}}_f = \frac{dg}{d\hat{\mathbf{U}}_f}. \quad (48 \text{ revisited})$$

Note that to obtain  $\hat{\mathbf{A}}_f$  one has to solve an additional system of equations. Subsequent substitution of Eq. (30) into Eq. (A.25) gives

$$d\mathcal{L} = \hat{\mathbf{A}}_f : \left( -\tilde{\mathbf{S}}_f^T d\tilde{\mathbf{K}}\hat{\mathbf{U}} + \tilde{\mathbf{S}}_f^T d\tilde{\mathbf{F}} + d\hat{\mathbf{F}}_f - \tilde{\mathbf{K}}_{fp}d\hat{\mathbf{U}}_p \right), \quad (A.28)$$

where we also used

$$\hat{\mathbf{U}} = \tilde{\mathbf{S}}_f\hat{\mathbf{U}}_f + \tilde{\mathbf{S}}_p\hat{\mathbf{U}}_p. \quad (A.29)$$

Note that Eq. (A.28) can be rewritten as

$$d\mathcal{L} = \left( -\tilde{\mathbf{S}}_f\hat{\mathbf{A}}_f \otimes \hat{\mathbf{U}} \right) : d\tilde{\mathbf{K}} + \tilde{\mathbf{S}}_f\hat{\mathbf{A}}_f : d\tilde{\mathbf{F}} + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f - \tilde{\mathbf{K}}_{pf}\hat{\mathbf{A}}_f : d\hat{\mathbf{U}}_p. \quad (A.30)$$

Substitution of Eqs. (35) and (A.14) into Eq. (A.30) yields

$$d\mathcal{L} = \left( \mathbf{A}\tilde{\mathbf{S}}_f\hat{\mathbf{A}}_f \otimes \mathbf{D} \right) : d\mathbf{K} - \tilde{\mathbf{X}}_f\tilde{\mathbf{S}}_f\hat{\mathbf{A}}_f : d\check{\mathbf{F}}_f + \mathbf{C}\tilde{\mathbf{S}}_f\hat{\mathbf{A}}_f : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f - \tilde{\mathbf{K}}_{pf}\hat{\mathbf{A}}_f : d\hat{\mathbf{U}}_p, \quad (A.31)$$

with

$$\mathbf{D} := \mathbf{B} - \mathbf{A}\hat{\mathbf{U}}. \quad (A.32)$$

Thus to obtain  $d\hat{\mathbf{U}}_f$  one requires an extra solve or matrix–vector product (depending on the self-adjointness), Eq. (48), per analysis based on the reduced system.

#### A.4. Sensitivities of the reaction load

From the solution of Eq. (27) the reaction loads can be calculated via

$$\hat{\mathbf{F}}_p = \tilde{\mathbf{K}}_{pf} \hat{\mathbf{U}}_f + \tilde{\mathbf{K}}_{pp} \hat{\mathbf{U}}_p - \tilde{\mathbf{F}}_p, \quad (28 \text{ revisited})$$

Note that here we have again omitted the subscript  $i$ . Sensitivities of a response function  $g \left[ \hat{\mathbf{F}}_p \left[ \tilde{\mathbf{K}}, \tilde{\mathbf{F}}, \hat{\mathbf{U}}_f, \hat{\mathbf{U}}_p \right] \right]$  are defined as

$$dg = \frac{dg}{d\hat{\mathbf{F}}_p} : d\hat{\mathbf{F}}_p. \quad (A.33)$$

To this end, we augment the response function with the linear system of equations from Eq. (27), which yields

$$\mathcal{L} \left[ \hat{\mathbf{F}}_p, \hat{\mathbf{A}}_f \right] := g \left[ \hat{\mathbf{F}}_p \right] - \hat{\mathbf{A}}_f : \left( \tilde{\mathbf{K}}_{ff} \hat{\mathbf{U}}_f - \tilde{\mathbf{F}}_f + \tilde{\mathbf{K}}_{fp} \hat{\mathbf{U}}_p - \tilde{\mathbf{F}}_f \right) \quad (A.34)$$

where  $\mathcal{L} \left[ \hat{\mathbf{F}}_p, \hat{\mathbf{A}}_f \right]$  is the Lagrangian and  $\hat{\mathbf{A}}_f \in \mathbb{R}^{\hat{f} \times l}$  the Lagrange multipliers related to  $\hat{\mathbf{U}}_f$ . Full differentiation of both sides gives

$$d\mathcal{L} = \frac{dg}{d\hat{\mathbf{F}}_p} : d\hat{\mathbf{F}}_p - \hat{\mathbf{A}}_f : \left( d\tilde{\mathbf{K}}_{ff} \hat{\mathbf{U}}_f + \tilde{\mathbf{K}}_{ff} d\hat{\mathbf{U}}_f - d\tilde{\mathbf{F}}_f + d\tilde{\mathbf{K}}_{fp} \hat{\mathbf{U}}_p + \tilde{\mathbf{K}}_{fp} d\hat{\mathbf{U}}_p - d\tilde{\mathbf{F}}_f \right), \quad (A.35)$$

with

$$d\hat{\mathbf{F}}_p = d\tilde{\mathbf{K}}_{pf} \hat{\mathbf{U}}_f + \tilde{\mathbf{K}}_{pf} d\hat{\mathbf{U}}_f + d\tilde{\mathbf{K}}_{pp} \hat{\mathbf{U}}_p + \tilde{\mathbf{K}}_{pp} d\hat{\mathbf{U}}_p - d\tilde{\mathbf{F}}_p. \quad (A.36)$$

Since  $d\hat{\mathbf{U}}_f$  is computationally expensive to calculate, aim is to circumvent this by a suitable choice of  $\hat{\mathbf{A}}_f$ . Collecting terms involving  $d\hat{\mathbf{U}}_f$  gives

$$\frac{dg}{d\hat{\mathbf{F}}_p} : \tilde{\mathbf{K}}_{pf} d\hat{\mathbf{U}}_f - \hat{\mathbf{A}}_f : \tilde{\mathbf{K}}_{ff} d\hat{\mathbf{U}}_f. \quad (A.37)$$

This can be rewritten<sup>11</sup> into

$$\left( \tilde{\mathbf{K}}_{fp} \frac{dg}{d\hat{\mathbf{F}}_p} - \tilde{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f \right) : d\hat{\mathbf{U}}_f. \quad (A.38)$$

To make the terms  $d\hat{\mathbf{U}}_f$  vanish, the term in parentheses must be zero. This can be achieved by choosing  $\hat{\mathbf{A}}_f$  as

$$\tilde{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f = \tilde{\mathbf{K}}_{fp} \frac{dg}{d\hat{\mathbf{F}}_p}. \quad (A.39)$$

Note that to obtain  $\hat{\mathbf{A}}_f$  one has to solve a system of equations. Substitution of Eqs. (30), (A.29) and (A.36) into Eq. (A.35) and rewriting yields

$$d\mathcal{L} = \frac{dg}{d\hat{\mathbf{F}}_p} : \left( \tilde{\mathbf{S}}_p^T d\tilde{\mathbf{K}} \hat{\mathbf{U}} + \tilde{\mathbf{K}}_{pp} d\hat{\mathbf{U}}_p - \tilde{\mathbf{S}}_p^T d\tilde{\mathbf{F}} \right) - \hat{\mathbf{A}}_f : \left( \tilde{\mathbf{S}}_f^T d\tilde{\mathbf{K}} \hat{\mathbf{U}} - d\tilde{\mathbf{F}}_f + \tilde{\mathbf{K}}_{fp} d\hat{\mathbf{U}}_p - \tilde{\mathbf{S}}_f^T d\tilde{\mathbf{F}} \right) \quad (A.40)$$

Subsequent substitution of Eqs. (35) and (A.14) and simplifying using Eqs. (A.13) and (A.32) yields

$$dg = (\mathbf{A}\mathbf{E} \otimes \mathbf{D}) : d\mathbf{K} - \check{\mathbf{X}}_f \mathbf{E} : d\check{\mathbf{F}}_f + \mathbf{C}\mathbf{E} : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f + \left( \tilde{\mathbf{K}}_{pp} \frac{dg}{d\hat{\mathbf{F}}_p} - \tilde{\mathbf{K}}_{pf} \hat{\mathbf{A}}_f \right) : d\hat{\mathbf{U}}_p, \quad (A.41)$$

with temporary<sup>12</sup> variable

$$\mathbf{E} := \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f - \tilde{\mathbf{S}}_p \frac{dg}{d\hat{\mathbf{F}}_p}. \quad (A.42)$$

<sup>12</sup> Only used within this section.

Thus to obtain  $\hat{\mathbf{dF}}_p$  one requires an extra solve or matrix–vector product (depending on the self-adjointness), Eq. (A.39), per analysis based on using the reduced system.

#### A.5. Sensitivities of the state on the secondary DOFs

After obtaining the solution to the state of the primary DOFs one can obtain the solution of the free secondary DOFs via

$$\check{\mathbf{U}}_f = - \left( \check{\mathbf{X}}_f \hat{\mathbf{U}} + \check{\mathbf{V}}_f \right). \quad (22 \text{ revisited})$$

We augment the response function with the linear systems of equations from Eqs. (20), (21) and (27), that is

$$\begin{aligned} \mathcal{L} \left[ \check{\mathbf{U}}_f, \check{\mathbf{A}}_f, \check{\mathbf{A}}_f, \hat{\mathbf{A}}_f \right] := g \left[ \check{\mathbf{U}}_f \right] &- \check{\mathbf{A}}_f : \left( \check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f - \check{\mathbf{K}}_f \right) \\ &- \check{\mathbf{A}}_f : \left( \check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f - \check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p + \check{\mathbf{F}}_f \right) \\ &- \hat{\mathbf{A}}_f : \left( \check{\mathbf{K}}_{ff} \hat{\mathbf{U}}_f - \hat{\mathbf{F}}_f + \check{\mathbf{K}}_{fp} \hat{\mathbf{U}}_p - \tilde{\mathbf{F}}_f \right). \end{aligned} \quad (A.43)$$

Full differentiation of both sides gives

$$\begin{aligned} d\mathcal{L} = - \frac{dg}{d\check{\mathbf{U}}_f} : \left( d\check{\mathbf{X}}_f \hat{\mathbf{U}} + \check{\mathbf{X}}_f d\hat{\mathbf{U}} + d\check{\mathbf{V}}_f \right) \\ - \check{\mathbf{A}}_f : \left( d\check{\mathbf{K}}_{ff} \check{\mathbf{X}}_f + \check{\mathbf{K}}_{ff} d\check{\mathbf{X}}_f - d\check{\mathbf{K}}_f \right) \\ - \check{\mathbf{A}}_f : \left( d\check{\mathbf{K}}_{ff} \check{\mathbf{V}}_f + \check{\mathbf{K}}_{ff} d\check{\mathbf{V}}_f - d\check{\mathbf{K}}_{fp} \check{\mathbf{U}}_p - \check{\mathbf{K}}_{fp} d\check{\mathbf{U}}_p + d\check{\mathbf{F}}_f \right) \\ - \hat{\mathbf{A}}_f : \left( d\check{\mathbf{K}}_{ff} \hat{\mathbf{U}}_f + \check{\mathbf{K}}_{ff} d\hat{\mathbf{U}}_f - d\hat{\mathbf{F}}_f + d\check{\mathbf{K}}_{fp} \hat{\mathbf{U}}_p + \check{\mathbf{K}}_{fp} d\hat{\mathbf{U}}_p - d\tilde{\mathbf{F}}_f \right). \end{aligned} \quad (A.44)$$

Since  $d\check{\mathbf{X}}_f$ ,  $d\check{\mathbf{V}}_f$  and  $d\hat{\mathbf{U}}_f$  are computationally expensive to calculate, aim is to circumvent this by a suitable choices of  $\check{\mathbf{A}}_f$ ,  $\check{\mathbf{A}}_f$  and  $\hat{\mathbf{A}}_f$ . Collecting terms involving  $d\hat{\mathbf{U}}_f$  gives

$$\left( -\check{\mathbf{S}}_f^T \check{\mathbf{X}}_f \frac{dg}{d\check{\mathbf{U}}_f} - \check{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f \right) : d\hat{\mathbf{U}}_f, \quad (A.45)$$

where we used

$$d\hat{\mathbf{U}} = \check{\mathbf{S}}_f d\hat{\mathbf{U}}_f + \check{\mathbf{S}}_p d\hat{\mathbf{U}}_p. \quad (A.46)$$

To make the terms  $d\hat{\mathbf{U}}_f$  vanish we define  $\hat{\mathbf{A}}_f$  as

$$\check{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f = -\check{\mathbf{S}}_f^T \check{\mathbf{X}}_f \frac{dg}{d\check{\mathbf{U}}_f}, \quad (A.47)$$

which requires an additional solve using the condensed system. Collecting terms involving  $d\check{\mathbf{V}}_f$  gives

$$\left( -\frac{dg}{d\check{\mathbf{U}}_f} - \check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f \right) : d\check{\mathbf{V}}_f. \quad (A.48)$$

To make the terms  $d\check{\mathbf{V}}_f$  vanish we define  $\check{\mathbf{A}}_f$  as

$$\check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f = -\frac{dg}{d\check{\mathbf{U}}_f}, \quad (A.49)$$

for which an additional full system solve is needed. Finally, collecting terms involving  $d\check{\mathbf{X}}_f$  gives

$$\left( -\frac{dg}{d\check{\mathbf{U}}_f} \hat{\mathbf{U}}^T - \check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f \right) : d\check{\mathbf{X}}_f. \quad (A.50)$$

To make the terms  $d\check{\mathbf{X}}_f$  vanish we define  $\check{\hat{\mathbf{A}}}_f$  as

$$\check{\mathbf{K}}\check{\hat{\mathbf{A}}}_f = -\frac{dg}{d\check{\mathbf{U}}_f}\hat{\mathbf{U}}^\top. \quad (\text{A.51})$$

Thus, with Eq. (A.49)

$$\check{\hat{\mathbf{A}}}_f[\check{\mathbf{A}}_f] = \check{\mathbf{A}}_f\hat{\mathbf{U}}^\top. \quad (\text{A.52})$$

With all multipliers defined, substitution of Eqs. (30), (A.46) and (A.52) into Eq. (A.44) and simplifying using Eq. (A.32) gives

$$\begin{aligned} d\mathcal{L} = & -\left(\check{\mathbf{S}}_f\check{\mathbf{A}}_f \otimes \mathbf{D}\right) : d\mathbf{K} - \left(\check{\mathbf{S}}_f\hat{\mathbf{A}}_f \otimes \hat{\mathbf{U}}\right) : d\check{\mathbf{K}} + \check{\mathbf{S}}_f\hat{\mathbf{A}}_f : d\tilde{\mathbf{F}} \\ & - \check{\mathbf{A}}_f : d\check{\mathbf{F}}_f + \check{\mathbf{K}}_{pf}\check{\mathbf{A}}_f : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f - \left(\check{\mathbf{K}}_{pf}\hat{\mathbf{A}}_f + \check{\mathbf{S}}_p^\top\check{\mathbf{X}}_f\frac{dg}{d\check{\mathbf{U}}_f}\right) : d\hat{\mathbf{U}}_p. \end{aligned} \quad (\text{A.53})$$

Substitution of Eqs. (35) and (A.14) into Eq. (A.53) yields

$$\begin{aligned} d\mathcal{L} = & \left[\left(\mathbf{A}\check{\mathbf{S}}_f\hat{\mathbf{A}}_f - \check{\mathbf{S}}_f\check{\mathbf{A}}_f\right) \otimes \mathbf{D}\right] : d\mathbf{K} \\ & - \left(\check{\mathbf{X}}_f\check{\mathbf{S}}_f\hat{\mathbf{A}}_f + \check{\mathbf{A}}_f\right) : d\check{\mathbf{F}}_f + \left(\mathbf{A}\check{\mathbf{S}}_f\hat{\mathbf{A}}_f + \check{\mathbf{K}}_{pf}\check{\mathbf{A}}_f\right) : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f - \left(\check{\mathbf{K}}_{pf}\hat{\mathbf{A}}_f + \check{\mathbf{S}}_p^\top\check{\mathbf{X}}_f\frac{dg}{d\check{\mathbf{U}}_f}\right) : d\hat{\mathbf{U}}_p. \end{aligned} \quad (\text{A.54})$$

#### A.6. Sensitivities of the reaction load on the secondary DOFs

Based on the solution of Eq. (22) one can also analyse the reaction loads of the secondary DOFs via

$$\check{\mathbf{F}}_p = \check{\mathbf{K}}_p\hat{\mathbf{U}} + \check{\mathbf{K}}_{pf}\check{\mathbf{U}}_f + \check{\mathbf{K}}_{pp}\check{\mathbf{U}}_p. \quad (\text{23 revisited})$$

We augment the response function with the linear systems of equations from Eqs. (20), (21) and (27), that is

$$\begin{aligned} \mathcal{L}[\check{\mathbf{F}}_p, \check{\mathbf{A}}_f, \check{\mathbf{A}}_f, \hat{\mathbf{A}}_f] := & g[\check{\mathbf{F}}_p] - \check{\mathbf{A}}_f : \left(\check{\mathbf{K}}_{ff}\check{\mathbf{X}}_f - \check{\mathbf{K}}_f\right) \\ & - \check{\mathbf{A}}_f : \left(\check{\mathbf{K}}_{ff}\check{\mathbf{V}}_f - \check{\mathbf{K}}_{fp}\check{\mathbf{U}}_p + \check{\mathbf{F}}_f\right) \\ & - \hat{\mathbf{A}}_f : \left(\check{\mathbf{K}}_{ff}\hat{\mathbf{U}}_f - \hat{\mathbf{F}}_f + \check{\mathbf{K}}_{fp}\hat{\mathbf{U}}_p - \tilde{\mathbf{F}}_f\right). \end{aligned} \quad (\text{A.55})$$

Full differentiation of both sides gives

$$\begin{aligned} d\mathcal{L} = & \frac{dg}{d\check{\mathbf{F}}_p} : \left(d\check{\mathbf{K}}_p\hat{\mathbf{U}} + d\check{\mathbf{K}}_{pf}\check{\mathbf{U}}_f + d\check{\mathbf{K}}_{pp}\check{\mathbf{U}}_p\right) \\ & + \frac{dg}{d\check{\mathbf{F}}_p} : \left(\check{\mathbf{K}}_p d\hat{\mathbf{U}} - \check{\mathbf{K}}_{pf}\left(d\check{\mathbf{X}}_f\hat{\mathbf{U}} + \check{\mathbf{X}}_f d\hat{\mathbf{U}} + d\check{\mathbf{V}}_f\right) + \check{\mathbf{K}}_{pp}d\check{\mathbf{U}}_p\right) \\ & - \check{\mathbf{A}}_f : \left(d\check{\mathbf{K}}_{ff}\check{\mathbf{X}}_f + \check{\mathbf{K}}_{ff}d\check{\mathbf{X}}_f - d\check{\mathbf{K}}_f\right) \\ & - \check{\mathbf{A}}_f : \left(d\check{\mathbf{K}}_{ff}\check{\mathbf{V}}_f + \check{\mathbf{K}}_{ff}d\check{\mathbf{V}}_f - d\check{\mathbf{K}}_{fp}\check{\mathbf{U}}_p - \check{\mathbf{K}}_{fp}d\check{\mathbf{U}}_p + d\check{\mathbf{F}}_f\right) \\ & - \hat{\mathbf{A}}_f : \left(d\check{\mathbf{K}}_{ff}\hat{\mathbf{U}}_f + \check{\mathbf{K}}_{ff}d\hat{\mathbf{U}}_f - d\hat{\mathbf{F}}_f + d\check{\mathbf{K}}_{fp}\hat{\mathbf{U}}_p + \check{\mathbf{K}}_{fp}d\hat{\mathbf{U}}_p - d\tilde{\mathbf{F}}_f\right). \end{aligned} \quad (\text{A.56})$$

Since  $d\check{\mathbf{X}}_f$ ,  $d\check{\mathbf{V}}_f$  and  $d\hat{\mathbf{U}}_f$  are expensive to calculate, aim is to circumvent this by a suitable choices of  $\check{\hat{\mathbf{A}}}_f$ ,  $\check{\mathbf{A}}_f$  and  $\hat{\mathbf{A}}_f$ . Using Eq. (A.46) and subsequently collecting terms involving  $d\hat{\mathbf{U}}_f$  gives

$$\left(-\check{\mathbf{S}}_f^\top\mathbf{C}^\top\frac{dg}{d\check{\mathbf{F}}_p} - \check{\mathbf{K}}_{ff}\hat{\mathbf{A}}_f\right) : d\hat{\mathbf{U}}_f. \quad (\text{A.57})$$

To make the terms  $d\hat{\mathbf{U}}_f$  vanish we define  $\hat{\mathbf{A}}_f$  as

$$\tilde{\mathbf{K}}_{ff} \hat{\mathbf{A}}_f = -\tilde{\mathbf{S}}_f^T \mathbf{C}^T \frac{dg}{d\hat{\mathbf{F}}_p}, \quad (\text{A.58})$$

which requires an additional solve using the condensed system. Collecting terms involving  $d\check{\mathbf{V}}_f$  gives

$$\left( -\check{\mathbf{K}}_{fp} \frac{dg}{d\check{\mathbf{F}}_p} - \check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f \right) : d\check{\mathbf{V}}_f. \quad (\text{A.59})$$

To make the terms  $d\check{\mathbf{V}}_f$  vanish we define  $\check{\mathbf{A}}_f$  as

$$\check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f = -\check{\mathbf{K}}_{fp} \frac{dg}{d\check{\mathbf{F}}_p}, \quad (\text{A.60})$$

for which an additional full system solve is needed. Finally, collecting terms involving  $d\check{\mathbf{X}}_f$  gives

$$\left( -\check{\mathbf{K}}_{fp} \frac{dg}{d\check{\mathbf{F}}_p} \hat{\mathbf{U}}^T - \check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f \right) : d\check{\mathbf{X}}_f. \quad (\text{A.61})$$

To make the terms  $d\check{\mathbf{X}}_f$  vanish we define  $\check{\mathbf{A}}_f$  as

$$\check{\mathbf{K}}_{ff} \check{\mathbf{A}}_f = -\check{\mathbf{K}}_{fp} \frac{dg}{d\check{\mathbf{F}}_p} \hat{\mathbf{U}}^T. \quad (\text{A.62})$$

Thus, again

$$\check{\mathbf{A}}_f [\check{\mathbf{A}}_f] = \check{\mathbf{A}}_f \hat{\mathbf{U}}^T. \quad (\text{A.52 revisited})$$

Costs are comparable to those for the case in [Appendix A.5](#). With all multipliers defined, substitution of Eqs. (30), (A.46) and (A.52) into Eq. (A.56) and simplifying using Eq. (A.32) gives

$$\begin{aligned} d\mathcal{L} = & - \left( \check{\mathbf{S}}_f \check{\mathbf{A}}_f \otimes \mathbf{D} \right) : d\mathbf{K} - \left( \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f \otimes \hat{\mathbf{U}} \right) : d\tilde{\mathbf{K}} + \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f : d\tilde{\mathbf{K}} \\ & - \check{\mathbf{A}}_f : d\check{\mathbf{F}}_f + \left( \check{\mathbf{K}}_{pp} \frac{dg}{d\check{\mathbf{F}}_p} + \check{\mathbf{K}}_{pf} \check{\mathbf{A}}_f \right) : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f - \left( \tilde{\mathbf{K}}_{pf} \hat{\mathbf{A}}_f + \tilde{\mathbf{S}}_p^T \mathbf{C}^T \frac{dg}{d\check{\mathbf{F}}_p} \right) : d\hat{\mathbf{U}}_p. \end{aligned} \quad (\text{A.63})$$

Substitution of Eqs. (35) and (A.14) into Eq. (A.63) yields

$$\begin{aligned} d\mathcal{L} = & \left[ \left( \check{\mathbf{S}}_f \check{\mathbf{A}}_f - \mathbf{A} \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f \right) \otimes \mathbf{D} \right] : d\mathbf{K} \\ & - \left( \check{\mathbf{X}}_f \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f + \check{\mathbf{A}}_f \right) : d\check{\mathbf{F}}_f + \left( \check{\mathbf{K}}_{pp} \frac{dg}{d\check{\mathbf{F}}_p} + \mathbf{C} \tilde{\mathbf{S}}_f \hat{\mathbf{A}}_f + \check{\mathbf{K}}_{pf} \check{\mathbf{A}}_f \right) : d\check{\mathbf{U}}_p + \hat{\mathbf{A}}_f : d\hat{\mathbf{F}}_f \\ & - \left( \tilde{\mathbf{K}}_{pf} \hat{\mathbf{A}}_f + \tilde{\mathbf{S}}_p^T \mathbf{C}^T \frac{dg}{d\check{\mathbf{F}}_p} \right) : d\hat{\mathbf{U}}_p. \end{aligned} \quad (\text{A.64})$$

### A.7. Summary

For clarity the resulting sensitivities as obtained in [Appendices A.1 to A.6](#) are summarized in [Table A.3](#). It also lists the additional solves, along with their dimensionality and number of right-hand-sides, required to calculate the sensitivities. Here we assumed a worst-case scenario, where a response  $g$  depends on *all* the  $l^{(i)}$  states  $\hat{\mathbf{U}}_f^{(i)}$ ,  $\hat{\mathbf{F}}_p^{(i)}$ ,  $\check{\mathbf{U}}_f^{(i)}$  and/or  $\check{\mathbf{F}}_p^{(i)}$  of analysis  $i$ . In practice the number of right-hand-sides for a single response depends on the dependency of the response with respect to each state. [Table A.3](#) shows expensive additional solves are only required for responses depending on the state(s) or reaction loads of secondary DOFs.

**Table A.3**

Sensitivities of common response dependencies with respect to the system matrix, applied loads and prescribed states. Format is  $dg_i [x_i] = \sum_j a_{ij} : db_j$ , with  $i$  the row and  $j$  the column. Note the dimensionality of  $\hat{\mathbf{A}}_f \in \mathbb{R}^{f(i) \times l(i)}$  and  $\hat{\mathbf{A}}_f \in \mathbb{R}^{f(i) \times l(i)}$ .

| $x$                          | $d\mathbf{K}$  | $d\mathbf{U}_p^{(i)}$   | $d\mathbf{F}_f^{(i)}$                                | $d\mathbf{U}_p^{(i)}$   | $d\mathbf{F}_f^{(i)}$      | $a$   | Solves  |
|------------------------------|--|---|--|---|----------------------------|---|---|
| $\tilde{\mathbf{K}}$         | $\mathbf{A}a \otimes \mathbf{A}$   |   |  |   |                            | $\frac{dg}{d\mathbf{K}}$  |   |
| $\tilde{\mathbf{F}}^{(i)}$   | $\mathbf{A}a \otimes \mathbf{B}^{(i)}$   | $\mathbf{C}a$   | $-\tilde{\mathbf{X}}_f a$                            |   |                            | $\frac{dg}{d\mathbf{F}^{(i)}}$  |   |
| $\hat{\mathbf{U}}_f^{(i)}$   | $\mathbf{A}a \otimes \mathbf{D}^{(i)}$   | $\mathbf{C}a$   | $-\tilde{\mathbf{X}}_f a$                            | $-\tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)}$   | $\hat{\mathbf{A}}_f^{(i)}$ | $\tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)}$   | $\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{A}}_f^{(i)} = \frac{dg}{d\mathbf{U}_f^{(i)}}$   |
| $\hat{\mathbf{F}}_p^{(i)}$   | $\mathbf{A}a \otimes \mathbf{D}^{(i)}$   | $\mathbf{C}a$   | $-\tilde{\mathbf{X}}_f a$                            | $-\tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)} + \tilde{\mathbf{K}}_{pp}^{(i)} \frac{dg}{d\mathbf{F}_p^{(i)}}$                    | $\hat{\mathbf{A}}_f^{(i)}$ | $\tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)} - \tilde{\mathbf{S}}_p^{(i)} \frac{dg}{d\mathbf{F}_p^{(i)}}$ | $\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{A}}_f^{(i)} = \tilde{\mathbf{K}}_{fp}^{(i)} \frac{dg}{d\mathbf{F}_p^{(i)}}$                     |
| $\tilde{\mathbf{U}}_f^{(i)}$ | $(\mathbf{A}a - \tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)}) \otimes \mathbf{D}^{(i)}$ | $\mathbf{C}a + \tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)}$  | $-\tilde{\mathbf{X}}_f a - \hat{\mathbf{A}}_f^{(i)}$ | $-\tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)} - \tilde{\mathbf{S}}_p^{(i)T} \tilde{\mathbf{X}}_f \frac{dg}{d\mathbf{U}_f^{(i)}}$ | $\hat{\mathbf{A}}_f^{(i)}$ | $\tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)}$   | $\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{A}}_f^{(i)} = -\tilde{\mathbf{S}}_f^{(i)T} \tilde{\mathbf{X}}_f \frac{dg}{d\mathbf{U}_f^{(i)}}$ |
| $\tilde{\mathbf{F}}_p^{(i)}$ | $(\mathbf{A}a - \tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)}) \otimes \mathbf{D}$       | $\mathbf{C}a + \tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)} + \tilde{\mathbf{K}}_{pp}^{(i)} \frac{dg}{d\mathbf{F}_p^{(i)}}$ | $-\tilde{\mathbf{X}}_f a - \hat{\mathbf{A}}_f^{(i)}$ | $-\tilde{\mathbf{K}}_{pf}^{(i)} \hat{\mathbf{A}}_f^{(i)} - \tilde{\mathbf{S}}_p^{(i)T} \mathbf{C}^T \frac{dg}{d\mathbf{F}_p^{(i)}}$         | $\hat{\mathbf{A}}_f^{(i)}$ | $\tilde{\mathbf{S}}_f^{(i)} \hat{\mathbf{A}}_f^{(i)}$   | $\tilde{\mathbf{K}}_{ff}^{(i)} \hat{\mathbf{A}}_f^{(i)} = -\tilde{\mathbf{S}}_f^{(i)T} \mathbf{C}^T \frac{dg}{d\mathbf{F}_p^{(i)}}$         |

## References

- [1] O. Amir, M. Bendsøe, O. Sigmund, Internat. J. Numer. Methods Engrg. 78 (12) (2009) 1474–1491, <http://dx.doi.org/10.1002/nme.2536>, arXiv:1010.1724.
- [2] T. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006.
- [3] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.
- [4] O. Amir, N. Aage, B. Lazarov, Struct. Multidiscip. Optim. 49 (5) (2014) 815–829, <http://dx.doi.org/10.1007/s00158-013-1015-5>.
- [5] G. Rozvany, O. Sigmund, T. Lewiński, D. Gerdes, T. Birker, Struct. Optim. 5 (3) (1993) 204–206, <http://dx.doi.org/10.1007/bf01743359>.
- [6] M. Cavazzuti, A. Baldini, E. Bertocchi, D. Costi, E. Torricelli, P. Moruzzi, Struct. Multidiscip. Optim. 44 (1) (2011) 45–56, <http://dx.doi.org/10.1007/s00158-010-0578-7>.
- [7] A.R. Diaz, M.P. Bendsøe, Struct. Optim. 4 (1) (1992) 17–22, <http://dx.doi.org/10.1007/BF01894077>.
- [8] X.S. Zhang, E. de Sturler, G.H. Paulino, Comput. Methods Appl. Mech. Engrg. 325 (2017) 463–487, <http://dx.doi.org/10.1016/j.cma.2017.06.035>.
- [9] X.S. Zhang, E. de Sturler, A. Shapiro, J. Appl. Mech. 87 (5) (2020) <http://dx.doi.org/10.1115/1.4045902>.
- [10] Z. Liu, J. Korvink, Finite Elem. Anal. Des. 45 (8–9) (2009) 555–568, <http://dx.doi.org/10.1016/j.finel.2009.03.005>.
- [11] M. Jin, X. Zhang, Z. Yang, B. Zhu, J. Mech. Des. 140 (1) (2017) 011402, <http://dx.doi.org/10.1115/1.4038332>.
- [12] A. Evgrafov, C. Rupp, K. Maute, M. Dunn, Struct. Multidiscip. Optim. 36 (4) (2008) 329–345, <http://dx.doi.org/10.1007/s00158-007-0190-7>.
- [13] N. Aage, E. Andreassen, B. Lazarov, Struct. Multidiscip. Optim. 51 (3) (2015) 565–572, <http://dx.doi.org/10.1007/s00158-014-1157-0>.
- [14] J. Wu, C. Dick, R. Westermann, IEEE Trans. Vis. Comput. Graphics 22 (3) (2016) 1195–1208, <http://dx.doi.org/10.1109/TVCG.2015.2502588>.
- [15] U. Kirsch, Struct. Optim. 6 (3) (1993) 143–150, <http://dx.doi.org/10.1007/BF01743505>.
- [16] O. Amir, Struct. Multidiscip. Optim. 51 (1) (2015) 41–57, <http://dx.doi.org/10.1007/s00158-014-1098-7>.
- [17] G.-H. Yoon, Comput. Methods Appl. Mech. Engrg. 199 (25–28) (2010) 1744–1763, <http://dx.doi.org/10.1016/j.cma.2010.02.002>.
- [18] Y. Choi, G. Oxberry, D. White, T. Kirchdoerfer, Accelerating design optimization using reduced order models, 2019, <http://dx.doi.org/10.13140/RG.2.2.16056.08965>, arXiv:1909.11320.
- [19] R. Guyan, AIAA J. 3 (2) (1965) 380, <http://dx.doi.org/10.2514/3.2874>.
- [20] B. Irons, AIAA J. 3 (5) (1965) 961–962, <http://dx.doi.org/10.2514/3.3027>.
- [21] J. Przemieniecki, AIAA J. 1 (1) (1963) 138–147.
- [22] L. Schmit, M. Hirokazu, Approximation concepts for efficient structural synthesis, Technical Report, NASA, 1976, pp. 1–314.
- [23] A. Noor, H. Kamel, R. Fulton, Comput. Struct. 8 (5) (1978) 621–632, [http://dx.doi.org/10.1016/0045-7949\(78\)90100-1](http://dx.doi.org/10.1016/0045-7949(78)90100-1).
- [24] G. Kron, J. Appl. Mech. 22 (2) (1955) 235–244.



- [25] E. Wilson, *Internat. J. Numer. Methods Engrg.* 8 (1) (1974) 198–203, <http://dx.doi.org/10.1002/nme.1620080115>.
- [26] S. Bencotter, *J. Appl. Mech.* 15 (4) (1948) 303–307.
- [27] E. Haynsworth, *On the schur complement*, 1968.
- [28] I. Schur, *J. Die Reine Und Angew. Math.* 147 (1917) 205–232.
- [29] M. Botkin, R. Yang, *AIAA J.* 29 (3) (1989) 486–488, <http://dx.doi.org/10.2514/3.59924>.
- [30] S. Gangadharan, R. Haftka, E. Nikolaidis, *Commun. Appl. Numer. Methods* 6 (3) (1990) 161–171, <http://dx.doi.org/10.1002/cnm.1630060302>.
- [31] R. Yang, C. Lu, *AIAA J.* 34 (7) (1996) 1533–1535, <http://dx.doi.org/10.2514/3.60028>.
- [32] G.-H. Yoon, Y.-S. Joung, Y.-Y. Kim, *Internat. J. Numer. Methods Engrg.* 69 (6) (2007) 1278–1304, <http://dx.doi.org/10.1002/nme.1808>.
- [33] L. Xia, J.-H. Zhu, W.-H. Zhang, *Struct. Multidiscip. Optim.* 45 (5) (2012) 643–655, <http://dx.doi.org/10.1007/s00158-011-0720-1>.
- [34] J. Groen, M. Langelaar, O. Sigmund, M. Ruess, *Internat. J. Numer. Methods Engrg.* 110 (10) (2017) 903–920, <http://dx.doi.org/10.1002/nme.5432>.
- [35] E. Amir, *Struct. Multidiscip. Optim.* 59 (1) (2019) 165–184, <http://dx.doi.org/10.1007/s00158-018-2058-4>.
- [36] W.-H. Zhang, S. Sun, *Internat. J. Numer. Methods Engrg.* 68 (9) (2006) 993–1011, <http://dx.doi.org/10.1002/nme.1743>.
- [37] Z. Wu, L. Xia, S. Wang, T. Shi, *Comput. Methods Appl. Mech. Engrg.* 345 (2019) 602–617, <http://dx.doi.org/10.1016/j.cma.2018.11.003>.
- [38] M.Y. Wang, *J. Mech. Robot.* 1 (2) (2009) 021011, <http://dx.doi.org/10.1115/1.3056476>.
- [39] A. Hasse, L. Campanile, *Compliant mechanisms with selective compliance*, 2009, <http://dx.doi.org/10.1088/0964-1726/18/11/115016>.
- [40] A. Hasse, M. Franz, K. Mauser, *Mech. Mach. Sci.*, 45, Springer, Cham, 2017, pp. 227–238, [http://dx.doi.org/10.1007/978-3-319-45387-3\\_20](http://dx.doi.org/10.1007/978-3-319-45387-3_20).
- [41] L. Li, X. Zhu, *Struct. Multidiscip. Optim.* 60 (4) (2019) 1475–1489, <http://dx.doi.org/10.1007/s00158-019-02278-8>.
- [42] R. Cook, D. Malkus, M. Plesha, R. Witt, *Concepts and Applications of Finite Element Analysis*, 4th ed., John Wiley & Sons, Ltd., 2001, pp. 40–42.
- [43] J. Arora, E. Haug, *AIAA J.* 17 (9) (1979) 970–974.
- [44] A. Belegundu, *J. Struct. Eng. (United States)* 112 (8) (1986) 1971–1976, [http://dx.doi.org/10.1061/\(ASCE\)0733-9445\(1986\)112:8\(1971\)](http://dx.doi.org/10.1061/(ASCE)0733-9445(1986)112:8(1971)).
- [45] G. Vanderplaats, *AIAA J.* 18 (11) (1980) 1406–1407.
- [46] M.P. Bendsøe, N. Kikuchi, *Comput. Methods Appl. Mech. Engrg.* 71 (1988) 197–224, [http://dx.doi.org/10.1016/0045-7825\(88\)90086-2](http://dx.doi.org/10.1016/0045-7825(88)90086-2).
- [47] M. Yano, J.D. Penn, G. Konidaris, A.T. Patera, *Math, Numerics, & Programming (for Mechanical Engineers)*, MIT, 2012, September.
- [48] S. Boyd, S.P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [49] M.R. Hestenes, E. Stiefel, et al., *Methods of Conjugate Gradients for Solving Linear Systems*, Vol. 49, NBS Washington, DC, 1952.
- [50] T. Bruns, D. Tortorelli, *Comput. Methods Appl. Mech. Engrg.* 190 (26–27) (2001) 3443–3459, [http://dx.doi.org/10.1016/S0045-7825\(00\)00278-4](http://dx.doi.org/10.1016/S0045-7825(00)00278-4).
- [51] O. Sigmund, P. Clausen, *Comput. Methods Appl. Mech. Engrg.* 196 (13–16) (2007) 1874–1889, <http://dx.doi.org/10.1016/j.cma.2006.09.021>.
- [52] M.P. Bendsøe, O. Sigmund, *Topology Optimization*, Springer Berlin Heidelberg, 2004, <http://dx.doi.org/10.1007/978-3-662-05086-6>.
- [53] U. Larsen, O. Sigmund, S. Bouwstra, *J. Microelectromech. Syst.* 6 (2) (1997) 99–106, <http://dx.doi.org/10.1109/84.585787>.