Archives of Computational Methods in Engineering (2021) 28:2371–2408
https://doi.org/10.1007/s11831-020-09458-6

ORIGINAL PAPER

A New Lighting on Analytical Discrete Sensitivities in the Context of IsoGeometric Shape Optimization

T. Hirschler[1,2] · R. Bouclier[3,4] · A. Duval[1] · T. Elguedj[1] · J. Morlier[4]

Received: 22 May 2020 / Accepted: 29 June 2020 / Published online: 7 September 2020

Abstract
Isogeometric shape optimization has been now studied for over a decade. This contribution aims at compiling the key ingredients within this promising framework, with a particular attention to sensitivity analysis. Based on all the researches related to isogeometric shape optimization, we present a global overview of the process which has emerged. The principal feature is the use of two refinement levels of the same geometry: a coarse level where the shape updates are imposed and a fine level where the analysis is performed. We explain how these two models interact during the optimization, and especially during the sensitivity analysis. We present new theoretical developments, algorithms, and quantitative results regarding the analytical calculation of discrete adjoint-based sensitivities. In order to highlight the versatility of this sensitivity analysis method, we perform eight benchmark optimization examples with different types of objective functions (compliance, displacement field, stress field, and natural frequencies), different types of isogeometric element (2D and 3D standard solids, and a Kirchhoff–Love shell), and different types of structural analysis (static and vibration). The numerical performances of the analytical sensitivities are compared with approximate sensitivities. The results in terms of accuracy and numerical cost make us believe that the presented method is a viable strategy to build a robust framework for shape optimization.

1 Introduction

Structural shape optimization has been one of the early application of IsoGeometric Analysis whose seminal paper is Hughes et al.[42]. Wall et al.[87] have rapidly highlighted its benefit for shape optimization because IGA uses models that combine an accurate geometrical description and great analysis capabilities. Indeed, IGA employs spline-based geometric models to perform the analysis. More precisely, IGA is a Finite Element Method that uses a spline model to describe the domain geometry but also to represent the numerical solution of the problem using the isoparametric paradigm[19, 42]. Even in its original version, IGA draws on advanced and well-known technologies coming from the field of Computer-Aided Design, as for instance NURBS models. Nowadays, a large panel of spline technologies (T-Splines, LR B-Splines, etc.) is available for simulation[28, 69]. The growing interest for IGA does not only come from the possibility of having models with high quality geometries. These spline functions have also shown great performances when it comes to numerical simulation, and especially an increased per-degree-of-freedom accuracy in comparison with standard FEM[29]. IGA achieved to tackle demanding problems and became a solution of choice in specific fields, as for example fluid-structure interaction, or biomedical application[45, 60, 64].

Shape optimization might be one of these fields that could be pushed forward through the use of the isogeometric principle; and it already started. The reason is quite straightforward. Shape optimization requires a suitable mix of an accurate geometric description and an efficient analysis model. Even more importantly, a close link between both the geometric and the analysis models is highly sought since they repeatedly communicate during the resolution. This is where current standard approaches for structural shape optimization, based on classical FE models, face some

✉ T. Hirschler
thibaut.hirschler@epfl.ch

[1] Univ Lyon, INSA-Lyon, CNRS, LaMCoS UMR 5259, 69621 Villeurbanne Cedex, France

[2] École Polytechnique Fédérale de Lausanne, MATH, MNS, 1015 Lausanne, Switzerland

[3] Univ Toulouse, INSA-Toulouse, IMT UMR CNRS 5219, 31077 Toulouse Cedex 04, France

[4] Univ Toulouse, ISAE Supaero-INSA-Mines Albi-UPS, CNRS UMR5312, Institut Clément Ader, 31055 Toulouse Cedex 04, France

difficulties. Numerical approaches for the shape optimization of structures are not new and several strategies have been presented in the eighties and early nineties[9, 13, 35, 41, 43, 71]. These developments have led to two principal classes of methods: the node-based approaches, and the CAD-based approaches. The node-based method uses the nodes of an analysis model (i.e. the finite element mesh) as design parameters. Conversely, the CAD-based method uses two different models of the same structure: a parametrized CAD model that describes the geometry, and a FE model to perform the structural analysis. Having these two separated models has shown great benefits[13] and has been preferred over node-based methods for quite some time[41]. However, a major drawback of the method has restricted its deployment in design offices. Indeed, it requires a close link to handle the delicate task of transferring the information between the design model and the analysis model. To this purpose, very specific program needs to be developed[10]. For complex structures, the link between the geometric model and the finite element mesh is far from straightforward. Also, the repeated mesh generations during the resolution burdens this optimization process. Thus, more recently, node-based methods regain interest and efficient approaches have been developed[27, 40, 54, 78]. In case of node-based optimization, the difficulties lie in the treatments of the large number of design parameters. Indeed, the number of nodes of a FEM mesh can be significant. Thus, adequate strategies should be put in place in order to process all the data. Special care (known as sensitivity filtering) is needed to exploit the results coming from the FE Analysis such that appropriate shape updates are imposed[7].

Using IGA, we now have models that are suitable for both shape modeling and simulation. This key feature has been shown by Wall et al.[87], and by the increasing number of papers dealing with IGA-based shape optimization[15, 23, 32, 38, 39, 46, 50, 56, 59, 65–67, 75, 83, 88, 91, 93]. It concerns not only structural shape optimization but also other fields as heat conduction[92], electromagnetics[20, 68], fluid mechanics[73], and many other optimization problems. A general procedure, which has been improved over the years, is commonly adopted[21, 89]. It is based on a multilevel design concept which consists in choosing different refinement levels of the same spline-based geometry to define both optimization and analysis spaces[38, 50, 67, 88]. Shape updates are represented by altering the spatial location of the control points, and in some case the weights[67, 75], on the coarse level. The finer level defines the analysis model and is set to ensure good quality of the numerical solution. The optimization and analysis refinement levels are independently chosen which provides a problem-adapted choice of the spaces.

This contribution undertakes to synthesize the previous research on isogeometric shape optimization of structures. We present a general formalism where each step is detailed: it goes from the setting and modeling of a shape optimization problem to its resolution. We deal with both theoretical and practical aspects. Especially, we compile several benchmark examples that can be of interest for further researches and the investigation of new approaches. We do not restrict to a specific context in order to highlight the generality of the presented framework. Indeed, several types of objective functions are considered. Moreover, we investigate the case of two common structural analyses: the static linear analysis but also the natural frequency analysis. Finally, we deal with the case of 2D and 3D solid isogeometric elements as introduced by Hughes et al.[42], and also the case of the isogeometric Kirchhoff–Love shell of Kiendl et al.[49]. Practical information, quantitative data and discussions about the results are provided. Furthermore, this contribution offers a new lighting on how analytical sensitivity is achievable in the context of isogeometric shape optimization. It extends the existing works regarding this issue, as for instance Fußeder et al.[32] and Qian[75]. We show how analytical discrete adjoint-based sensitivity can be computed. All the examples tackled in this work are solved using this analytical method for the sensitivity analysis. Thus, it applies to standard solid elements but also for shells. In addition, it is not limited to the case of the compliance, but it can be used for a lot of response functions as for example stress-based criteria. We provide algorithms in order to illustrate how it can be efficiently implemented. For each numerical example, quantitative results on the sensitivities are given which is, to the author's knowledge and to a certain extent, missing in the literature. The performance of this new sensitivity analysis is compared with approximate calculations. It can be quite often red in the literature that analytical sensitivities are difficult to implement. This work offers the necessary ingredients to overcome it and shows its efficiency in term of computational time.

This new framework for isogeometric shape optimization of structures is presented as follows. Firstly, we present in Sect. 2 generalities about structural optimization and we describe the structural design problems that will be tackled in this work. Then, in Sect. 3, the different steps for the sensitivity analysis in IGA-based shape optimization are explained. We illustrate how information are shared from the design variable level to the analysis model and vice versa. Section 4 contains the main theoretical developments. We present how the principal required derivatives of the sensitivity analysis can be analytically computed. As already said, both standard solid and shell elements are considered. Finally, the results of the optimization problems are presented and discussed in Sect. 5. Concluding remarks on our observations and findings are given in Sect. 6.

2 IsoGeometric Shape Optimization

2.1 Structural Optimization

General Mathematical Formulation The mathematical formulation of an optimization problem involves different quantities. The objective function f quantifies the performance of the studied system. It is usually formulated such that the best solution is the one which returns the smallest value assessed by function f. The objective depends on several characteristics of the system, called variables or unknowns. In the specific case of structural optimization, we set design parameters that vary the geometry of the structure. We denote these parameters as design variables and we represent them as a vector of unknowns x. In this work, we only consider the case of continuum variables. Finally, we describe the space in which we are looking for the optimal solution through a combination of constraints c_i. Mathematically, the constraints are defined as scalar functions of x, and commonly takes the form of implicit equations or inequalities. With these notations in hand, an optimization problem can be formulated as follows:

$$\begin{aligned} &\text{minimize } f(x) \\ &\quad\text{w.r.t. } x \in \mathbb{R}^n \\ &\text{subjected to } c_i(x) = 0, \ \forall i \in \mathscr{E} \\ &\qquad\qquad c_i(x) \leq 0, \ \forall i \in \mathscr{I} \end{aligned} \tag{1}$$

where $\mathscr{E}$ and $\mathscr{I}$ are set of indices for equality and inequality constraints, respectively.

Structural Design In structural optimization, the goal is to improve the mechanical behavior of the structure. Thus, the expression of the objective function and/or the constraints involve quantities that describe the behavior of the structure. For instance, in the case of the static analysis of structures, the objective function is generally expressed as an explicit function of the design variables x and of the displacement field u. Moreover, in the context of computational mechanics, the analysis is performed through an approximated method, such as FEM, for example. In this work, we consider the discrete approach for shape optimization which means that the discretization step happens prior to the formulation of the optimization problem[85]. In this case, the objective function is expressed using the state variables $\boldsymbol{u}$:

$$f := f\big(x, \boldsymbol{u}(x)\big), \tag{2}$$

In the case of static linear analysis, the variables $\boldsymbol{u}$ (namely the displacement Degrees Of Freedom) implicitly depends on the design variables x through a linear system of equation:

$$\mathbf{K}(x)\boldsymbol{u} = \boldsymbol{F}(x). \tag{3}$$

Indeed, the stiffness matrix $\mathbf{K}$ and the load vector $\boldsymbol{F}$ are built through domain integrals. It means that their expressions depend on the shape of the structure, and consequently on the design variables x.

Let us also mention the case of natural frequency analysis which is also widely encountered in computational structural analysis. The discretization step results in an eigenvalue problem:

$$\big(\mathbf{K}(x) - \lambda\mathbf{M}(x)\big)\boldsymbol{v} = \mathbf{0}, \tag{4}$$

where $\mathbf{M}$ denotes the mass matrix. As before, the governing equations link implicitly the eigenvalues λ and their corresponding eigenvectors $\boldsymbol{v}$ to the design variables. Finally, the objective function may involve the aforementioned quantities:

$$f := f\big(x, \lambda(x), \boldsymbol{v}(x)\big). \tag{5}$$

In this work, we undertake to present a global framework that is not limited to one type of analysis. In the numerical experiments section, we will consider the case of static linear analysis as well as structural vibration analysis.

Optimization Algorithm There are numerous algorithms that enable to solve constrained optimization problems of form (1). Each algorithm is usually designed into a specific framework. In case of structural shape optimization, *gradient-based* algorithms are often used. This is possible when the objective and constraint functions are differentiable *w.r.t.* the design variables. For large problems with hundreds to thousands design variables, the use of gradients is quasi-inevitable to build an algorithm that converges in an acceptable amount of time. The information brought back by the gradients enables the algorithm to make suitable decisions during the resolution[72]. For large scale problems, gradient-free algorithm may require much more evaluation of the objective and constraint functions which drastically increase the overall computational time.

The computation of the sensitivities is a key step of the resolution for any *gradient-based* algorithm. This work deals with this issue in the context of isogeometric shape optimization. We show how full analytical sensitivities can be achieved for several objective functions and several types of element formulation (2D and 3D solid, and the widely used Kirchhoff–Love shell).

2.2 Targeted Numerical Examples

Let us describe as of now the optimization problems that we will consider in this work. Through the examples, we alternatively switch between standard IGA solid elements (2D or 3D) and a Kirchhoff–Love shell element. For all the numerical examples, we will focus on the computation of the sensitivities. Furthermore, this work aims at motivating the use of isogeometric analysis for shape optimization. Some of the presented examples have not yet been presented in the context of isogeometric shape optimization, and thus enlarge its scope of application.

Compliance Taking the compliance as the objective function is the most common choice in structural optimization. It can be expressed as follows:

$$f_c = \frac{1}{2}\boldsymbol{F} \cdot \boldsymbol{u}. \tag{6}$$

We will tackle three optimization problems that consists in minimizing the compliance under a given volume constraint:

- Plate with a hole (2D solid),
- Square roof (shell),
- 3D beam (3D solid).

Displacement Another possible choice for the objective function concerns the minimization of the displacement at a prescribed location (e.g. at point $\boldsymbol{M}$). Such an objective function can be expressed as follows:

$$f_u = \sqrt{\mathbf{u}^h \cdot \mathbf{u}^h}, \quad \text{with} \quad \mathbf{u}^h(\boldsymbol{M}) = \sum_a R_a(\boldsymbol{M})\boldsymbol{u}_a, \tag{7}$$

and where R_k denote some basis functions associated to the discretization. Instead of considering the displacement at a specific location, one may want to minimize the maximal deflection of a structure. It is known that solving min-max problems can be difficult due to non-differentiability of the max function[33, 58]. The discontinuity can be avoided by replacing the max function by an alternative continuous function[3, 86]. In this work, we employ the P-norm:

$$\boldsymbol{\Phi}_f = \left(\sum_{k=1}^{n_{pts}} |f_k|^P \right)^{1/P}. \tag{8}$$

where P is a positive integer. As P gets large, the function $\boldsymbol{\Phi}_f$ approaches the maximal value returned by function f at the n_{pts} selected locations. There exists alternative choices of aggregation functions as for instance the Kreisselmeier-Steinhauser function[3, 17, 86]. We will perform in this work two examples with, respectively, f_u and $\boldsymbol{\Phi}_u$ as the objective functions:

- 2D cantilever beam (2D solid),
- Square roof (shell).

Stress Field In order to prevent the failure of a structure, one may seek to reduce the maximal stresses due to, for example, stress concentration. Even if taking the compliance as an objective function tends to reduce the overall magnitude of the stress field, local stress with increased magnitude can appear[24, 97]. Adding the maximal stress into an optimization problem raises several difficulties due to its local nature[3, 86]. As for the maximal deflection, one solution consists in using an aggregation function that measures the maximal stress. In this work, this is done through the P-norm:

$$\boldsymbol{\Phi}_\sigma = \left(\sum_{k=1}^{n_{pts}} |\sigma_k|^P \right)^{1/P}. \tag{9}$$

We will perform in this work two examples with $\boldsymbol{\Phi}_\sigma$ as the objective function:

- 2D fillet (2D solid),
- Catenary arch (shell).

For the catenary arch problem, the objective will be to minimize the bending moment along the arch. For the fillet problem, the objective will be to minimize the Von-Mises stresses in the structure.

Natural Frequencies We will deal with a last objective function. One possibility we address here is to maximize the lower natural frequency λ_1 of the structure. This is done through the minimization of the following objective function:

$$f_{\lambda_1} = \frac{1}{\lambda_1}. \tag{10}$$

However, such an objective function may not be differentiable at specific configurations due to mode switching[66, 76, 96]. In this case, the algorithm faces difficulties to reach convergence. In order to define an objective function that is differentiable, one possibility is to aggregate (as for the stress-based optimization; see Eq. (9)) a group of n_λ frequencies in which the mode switching occurs:

$$\boldsymbol{\Phi}_\lambda = \left(\sum_{k=1}^{n_\lambda} \left(\frac{1}{\lambda_k} \right)^P \right)^{1/P}. \tag{11}$$

We will present one optimization example with $\boldsymbol{\Phi}_\lambda$ as the objective function:

- Elephant trunk (3D solid).

The development of the analytical sensitivities will be done in the context of static analysis. We will only mention the case of natural frequency analysis when dealing with this last optimization problem. However, we will see that the computation of the sensitivities is very similar in both contexts and involves the same steps and quantities. Again, we present this example in order to highlight the generality of the present framework.

2.3 A Multi-level Approach

The present framework relies on researches dealing with isogeometric shape optimization. A general procedure, which has been improved over the years is commonly adopted[21, 89]. The key feature and asset of isogeometric shape optimization relies on the possibility to properly choose both optimization and analysis spaces[32, 50, 65–67, 75]. A fine discretization is introduced as the analysis model in order to ensure good quality computations. Conversely, the optimization model (also called the design model) is defined to impose suitable shape variations. Both spaces describe the exact same geometry and are initially obtained through different refinement levels of the same geometric model[38].

During the optimization process, both models interact successively. Consequently, it is straightforward that these two models are also involved during the sensitivity analysis. Figure 1 illustrates the role of these models and how they communicate during the optimization process. In what follows, we present how each step depicted in Fig. 1 is formulated. It starts with the definition of a shape parametrization that links the design variables to the design model. Then, we need to enlighten the link between the design model and the analysis model: it concerns not only the shape update but also the sensitivity propagation. These first two parts are explained in Sect. 3. The main step of the sensitivity analysis occurs on the analysis model where *gradients* are computed. We deal with this issue in Sect. 4.

3 Shape Update and Sensitivity Propagation

3.1 Adjoint-Based Discrete Sensitivities

In this work, we perform adjoint-based discrete sensitivities. The starting point is the differentiation of the response function of the form (2). The total derivative of function f *w.r.t.* a design variable x_i reads as:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \frac{d\boldsymbol{u}}{dx_i}. \tag{12}$$

The term $d\boldsymbol{u}/dx_i$ is not explicitly known. We can identify this term by differentiating the discrete state Eqs. (3). It reads as:

$$\frac{d\boldsymbol{u}}{dx_i} = \mathbf{K}^{-1}\left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}\right). \tag{13}$$

Then, we can substitute Eq. (13) into Eq. (12):

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \mathbf{K}^{-1}\left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}\right). \tag{14}$$

One can see that the inverse $\mathbf{K}^{-1}$ is involved in the expression of the derivative. To a certain extent, it means that a resolution is required. There are two ways of dealing with this issue, namely the adjoint and the direct approaches. In the adjoint method, one solves firstly the adjoint problem:

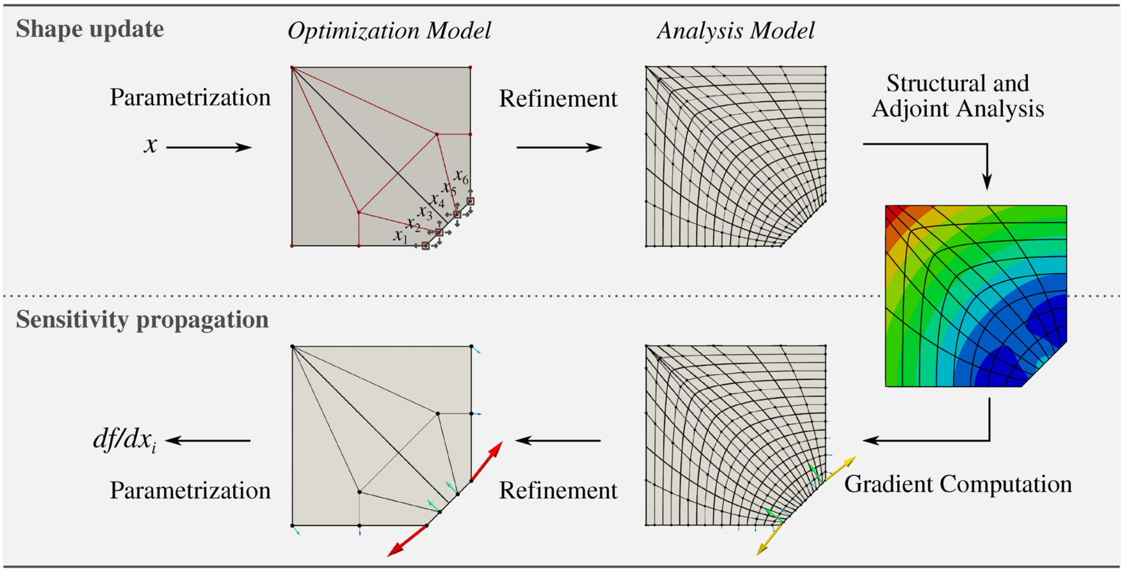

Fig. 1 Computation steps during the sensitivity analysis: the design variables act on the optimization model. The analysis model is a refined version of the optimization model (representing the exact same geometry) where the structural analysis is performed. This fine model also enables to compute the sensitivities. By recalling the refinement level and the definition of the shape parametrization, the sensitivities are propagated back to the design variable level

$$\mathbf{K}\boldsymbol{u}^* = \frac{\partial f}{\partial \boldsymbol{u}}. \tag{15}$$

The adjoint solution does not depend on the design variables. Consequently, for each design variable, the adjoint solution $\boldsymbol{u}^*$ is reused. Finally, the complete expression of the total derivative reads as follows:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u} \right). \tag{16}$$

Alternatively to the adjoint method, one can adopt the direct approach. It consists in solving several linear systems with the so called pseudo-load vectors as right-hand sides:

$$\mathbf{K}\boldsymbol{v}_i = \frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}. \tag{17}$$

Once the solution $\boldsymbol{v}_i$ is computed, one can get the total derivative:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \boldsymbol{v}_i. \tag{18}$$

Thus, in the direct sensitivity analysis there are potentially as many systems to be solved as design variables. Thus, adjoint sensitivity analysis is often preferred as long as the number of response functions involved in the optimization problems (objective and constraints) is smaller than the number of design variables.

One can see that, either in the direct or the adjoint approach, the derivatives of the stiffness matrix and of the load vector *w.r.t.* the design variables are involved. However the quantities $\mathbf{K}$ and $\boldsymbol{F}$ are defined on the analysis model, while the design variables act on the optimization model. In other words, there are intermediary steps that separate the design variables from the element operators. The idea to compute the aforementioned derivatives consists in applying several chain rules of differentiation accordingly to the steps that link the design variables to the element operators (i.e. the steps taking place during the shape update, see again Fig. 1).

3.2 Shape Representation and Parametrization

The first step is the *shape parametrization* that defines the design variables. In fact, until now, we did not clearly define what are these design variables.

NURBS Geometric Modeling In the context of isogeometric shape optimization, the whole process is based on geometric models. On the one hand these geometric models enable to represent the shape of the structure. On the other hand, these models are also used to perform the analysis. Historically, isogeometric analysis has been introduced using NURBS models (see Hughes et al.[42]), but it is not restricted to these geometric models. In this work, we only use NURBS models but it can surely be extended to other types of analysis-suitable geometric models.

There exists a large literature on NURBS modeling and related geometric modeling techniques. The interested reader can refer to Cohen et al.[16], Cottrell et al.[19], Farin[30] and Piegl and Tiller[74], to name just a few. Let us only introduce here the very basics. NURBS is the acronym for Non-Uniform Rational Basis Spline. It constitutes the today most commonly used technology in CAD. It describes complex geometric objects in the parametric form through the use of piecewise rational functions (see for example Piegl and Tiller[74] for more information). More precisely, a NURBS surface $\boldsymbol{S}$ and a NURBS volume $\boldsymbol{V}$ in the general 3D space are multivariate vector-valued functions of the form:

$$\boldsymbol{S}(\theta_1, \theta_2) = \sum_{k=1}^{n_{cp}} R_k(\theta_1, \theta_2)\boldsymbol{P}_k, \tag{19}$$

$$\boldsymbol{V}(\theta_1, \theta_2, \theta_3) = \sum_{k=1}^{n_{cp}} R_k(\theta_1, \theta_2, \theta_3)\boldsymbol{P}_k. \tag{20}$$

The parameters θ_i take real values in closed intervals (which form the parameter space $\bar{\Omega}$), usually [0, 1]. Each control point $\boldsymbol{P}_k$ is associated to a multivariate rational basis function R_k. These multivariate rational basis functions are built by tensor products and the weighting of univariate piecewise polynomial basis functions (the B-Spline functions). Finally, these univariate basis functions are defined by setting a polynomial degree and a knot vector. An example of a NURBS surface is given in Fig. 2. This surface represents one quarter of a square plate with a hole. The main input is the control net formed by the linear interpolation of the control points. In order to impose shape variations, an adequate choice consists in modifying the control point coordinates. By moving the location of the control points we alter the shape of the structure. This is a common choice in IGA-based shape optimization which takes up the idea behind CAD-based shape optimization (see for example Braibant and Fleury[13], Hsu[41] and Imam[43]). Due to the weighting occurring in the NURBS formulation, it is also possible to act on the weights associated to the control points to impose shape modifications[65–67, 75]. This option is not investigated in this work.

Shape Parametrization There is an infinite number of possibilities regarding the definition of the shape parametrization. One very simple parametrization lies in defining one independent design variable by movable control point. Each variable moves its associated control point in a specific direction:

$$\boldsymbol{P}_k = \boldsymbol{P}_k^0 + x_k\boldsymbol{n}_k. \tag{21}$$

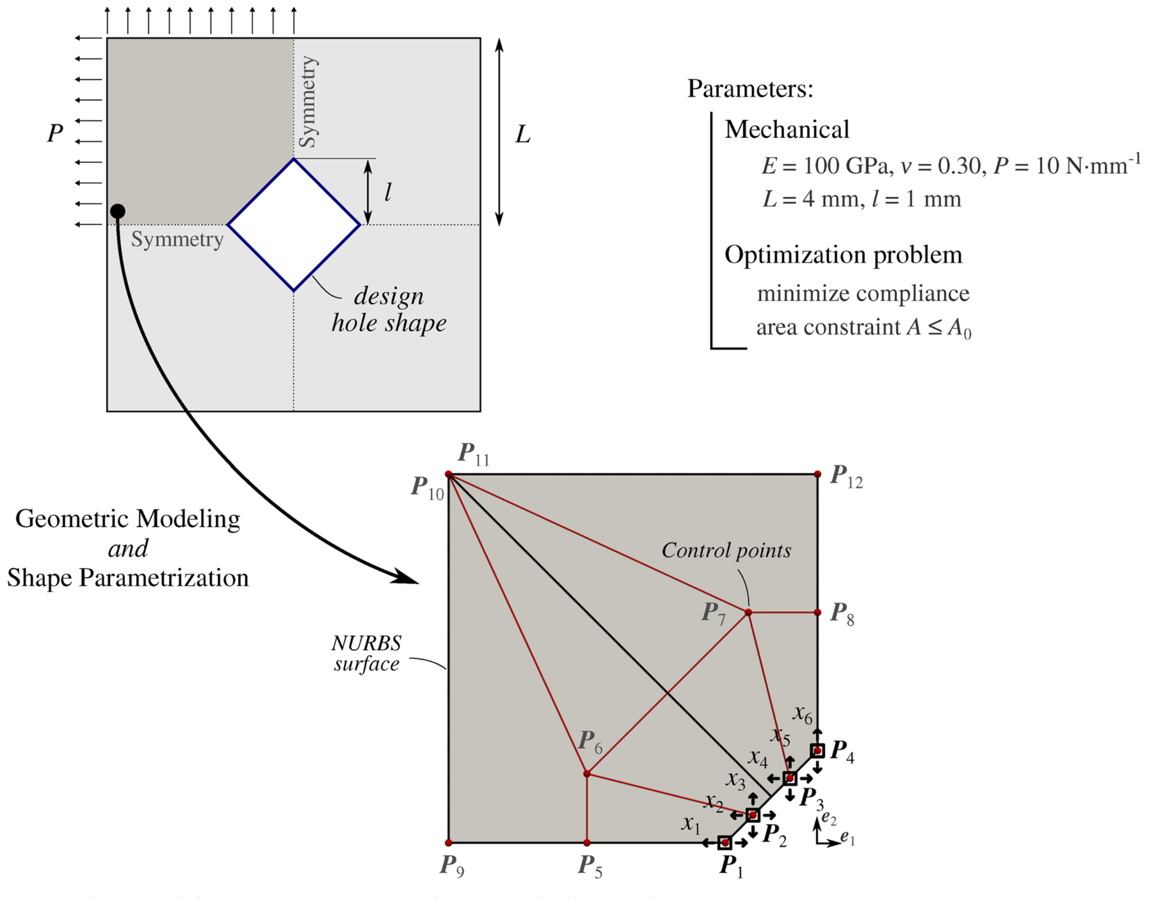

Fig. 2 NURBS modeling and shape parametrization for the plate with a hole problem

Identically, several design variables can be associated to a single control point. For example, one variable moves the control point in the x-direction and the second variable in direction y. Furthermore, it can be interesting to link one specific variable to multiple control points in order, for example, to preserve a geometric continuity or a symmetry. This more general shape parametrization takes the form:

$$\boldsymbol{P}_k = \boldsymbol{P}_k^0 + \sum_{i \in \mathcal{D}_k} x_i \boldsymbol{n}_i^k \,, \tag{22}$$

where $\mathcal{D}_k$ is the set of design variables acting on the kth control point. An example of design parametrization is given in Fig. 2 for the problem of a plate with a hole. Mathematically, this shape parametrization reads as:

$$\begin{aligned} \boldsymbol{P}_1 &= \boldsymbol{P}_1^0 + x_1\boldsymbol{e}_1 \\ \boldsymbol{P}_2 &= \boldsymbol{P}_2^0 + x_2\boldsymbol{e}_1 + x_3\boldsymbol{e}_2 \\ \boldsymbol{P}_3 &= \boldsymbol{P}_3^0 + x_4\boldsymbol{e}_1 + x_5\boldsymbol{e}_2 \\ \boldsymbol{P}_4 &= \boldsymbol{P}_4^0 + x_6\boldsymbol{e}_2 \\ \boldsymbol{P}_k &= \boldsymbol{P}_k^0 \quad \text{for } k > 4, \end{aligned} \tag{23}$$

where $\boldsymbol{e}_i$ are the Cartesian basis vectors. More complex shape parametrizations than those of the form (22) can be formulated, as for example to rotate a group of control points along an axis, etc.

The point to emphasize is that the shape parametrization links a group of design variables to the control point coordinates. Regarding the sensitivities, the differentiation of this shape parametrization will be involved. To highlight this point, let us denote by P_{jk} the j^{th} Cartesian component of the control point $\boldsymbol{P}_k$. With these notations, we can express the derivatives *w.r.t.* the design variables through the derivatives *w.r.t.* the control points of the optimization model:

$$\frac{\partial \bullet}{\partial x_i} = \sum_{k=1}^{n_{cp}} \sum_{j=1}^{3} \frac{\partial P_{jk}}{\partial x_i} \frac{\partial \bullet}{\partial P_{jk}} = \frac{\partial \boldsymbol{P}}{\partial x_i} : \frac{\partial \bullet}{\partial \boldsymbol{P}}, \tag{24}$$

where the symbol • denotes either a component of the stiffness matrix or of the load vector (or any quantity to be derived). The term $\partial \boldsymbol{P}/\partial x_i$ represents the differentiation of the shape parametrization. It contains the derivatives of the control points *w.r.t.* the design variables. In case of the shape parametrization of the form (22), this operator reads as:

$$\frac{\partial \boldsymbol{P}}{\partial x_i} = \left(\boldsymbol{n}_i^1, \boldsymbol{n}_i^2, \ldots, \boldsymbol{n}_i^{n_{cp}}\right)^T. \tag{25}$$

In what follows, we will see this operator as a matrix with size $n_{\mathrm{cp}} \times 3$. Furthermore, this operator is very sparse and one should take advantage of its sparsity for numerical efficiency. Indeed, one design variable usually moves only few control points.

For the example of the plate with a hole (see Fig. 2 and Eq. (23)), the derivatives of the shape parametrization are:

$$\frac{\partial \boldsymbol{P}}{\partial x_1} = \left(\boldsymbol{e}_1, 0, 0, 0, \ldots\right)^T \quad \frac{\partial \boldsymbol{P}}{\partial x_4} = \left(0, 0, \boldsymbol{e}_1, 0, \ldots\right)^T$$
$$\frac{\partial \boldsymbol{P}}{\partial x_2} = \left(0, \boldsymbol{e}_1, 0, 0, \ldots\right)^T \quad \frac{\partial \boldsymbol{P}}{\partial x_5} = \left(0, 0, \boldsymbol{e}_2, 0, \ldots\right)^T$$
$$\frac{\partial \boldsymbol{P}}{\partial x_3} = \left(0, \boldsymbol{e}_2, 0, 0, \ldots\right)^T \quad \frac{\partial \boldsymbol{P}}{\partial x_6} = \left(0, 0, 0, \boldsymbol{e}_2, \ldots\right)^T$$

where the dots symbol (…) indicates that the remaining components are filled with zeros. They correspond to the control points that are fixed.

3.3 From the Design Model to the Analysis Model

Spline Refinement As presented in Sect. 2.3, a major benefit of IGA-based shape optimization lies in the multilevel approach where the analysis model is a refined version of the design model. This is made possible by the refinement strategies that are available with B-Splines and NURBS. The first possibility consists in inserting new knots in the knot vectors. This process is called *knot insertion* and leads to increase the number of elements in the discretization, as shown in Fig. 3. It is also possible to elevate the degree of the underlying basis functions. Degree elevation keeps the same element density and the same regularity at the knots but increases the number of control points (and thus enriches the approximation space). Generally, these two refinement strategies are combined which offers great flexibility regarding the construction of the analysis model. This leads to the so called *k*-refinement[19] in which both the degree, regularity and element density are increased. For example in Fig. 3, the degree is elevated up to degree three in both direction and the number of elements is increased such that the final surface counts 4 × 4 elements. We refer the interested reader to Cottrell et al.[18, 19] and Hughes et al.[42] for more information on spline refinement.

Interestingly, these refinement procedures take the form of a linear application. The control points $\boldsymbol{Q}$ of the finer model are obtained through a linear relation of the form:

$$\boldsymbol{Q}^w = \mathbf{R}\boldsymbol{P}^w. \tag{26}$$

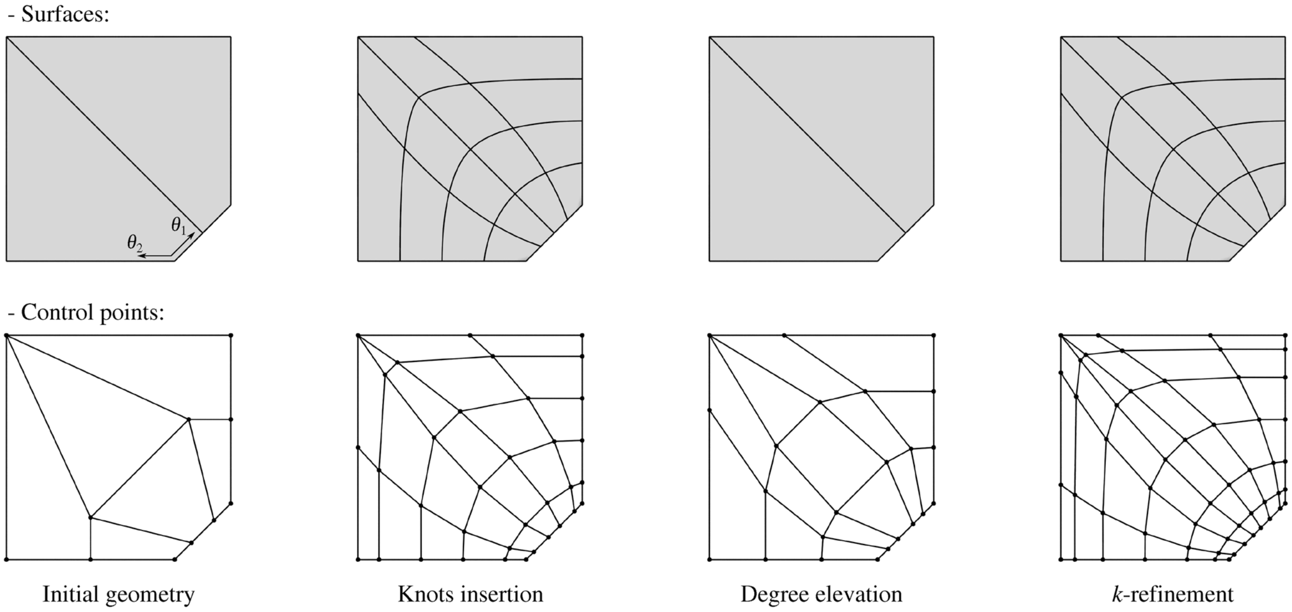

Fig. 3 Refinement strategies available with B-Splines and NURBS. These refinement procedure enables to generate the analysis model from the design model in IGA-based shape optimization

The refinement matrix $\mathbf{R}$ is a sparse rectangular matrix of size $m_{cp} \times n_{cp}$, m_{cp} being the number of control points of the analysis model. Equation (26) involves homogeneous coordinates: considering $\boldsymbol{P}_i = (x_i, y_i, z_i)$ with associated weight w_i, then $\boldsymbol{P}_i^w = (x_i w_i, y_i w_i, z_i w_i)$. In other words, we have:

$$\boldsymbol{P}^w = \text{diag}(\boldsymbol{w}_P)\boldsymbol{P} \tag{27}$$

where $\text{diag}(\boldsymbol{w}_P)$ is a diagonal matrix whose diagonal contains the n_{cp} weights associated to the control points $\boldsymbol{P}$. The expression of the refinement matrices in case of the knot insertion and degree elevation can be found in Lee and Park[55] and Piegl and Tiller[74], for example. The degree elevation is done in three steps: Bézier decomposition (knot insertion), degree elevation on each Bézier segment, and combination of the refined Bézier segments (knot removal). These three steps can be combined in order to form the refinement matrix $\mathbf{R}$.

Sensitivity Propagation The transition from the design model to the analysis model will also be involved during the sensitivity analysis. Indeed, after the introduction of the shape parametrization, we need to differentiate the stiffness matrix and the load vector *w.r.t.* the control point coordinates of the optimization model (i.e. the term $\partial\bullet/\partial\boldsymbol{P}$ in Eq. (24)). However, these operators are built using the analysis model and not using the design model. Therefore, to compute these derivatives we express them in terms of derivatives *w.r.t.* the control points of the analysis model. To that purpose, we operate once again a chain rule:

$$\frac{\partial\bullet}{\partial\boldsymbol{P}} = \frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{P}}\frac{\partial\bullet}{\partial\boldsymbol{Q}}. \tag{28}$$

The link between the control points $\boldsymbol{P}$ of the optimization model and the control points $\boldsymbol{Q}$ of the analysis model is given by the refinement matrix $\mathbf{R}$. More precisely, Eq. (26) enables us to write:

$$\frac{\partial\boldsymbol{Q}^w}{\partial\boldsymbol{P}^w} = \mathbf{R}^T. \tag{29}$$

Thus, the chain rule (28) takes the form:

$$\frac{\partial\bullet}{\partial\boldsymbol{P}^w} = \mathbf{R}^T\frac{\partial\bullet}{\partial\boldsymbol{Q}^w}. \tag{30}$$

The gradients $\partial\bullet/\partial\boldsymbol{P}^w$ and $\partial\bullet/\partial\boldsymbol{Q}^w$ should have appropriate shapes so that Eq. (30) makes sense. We suggest to view them as column matrices with respective sizes $n_{cp} \times 3$ and $m_{cp} \times 3$. Recalling Eqs. (27), (28), and (30), we get the following link between the derivatives *w.r.t.* the control points of the analysis and design models:

$$\frac{\partial\bullet}{\partial\boldsymbol{P}} = \left[\text{diag}(\boldsymbol{w}_P)\mathbf{R}^T\text{diag}(\boldsymbol{w}_Q)^{-1}\right]\frac{\partial\bullet}{\partial\boldsymbol{Q}}. \tag{31}$$

In the rest of the document, we omit the weighting terms ($\text{diag}(\boldsymbol{w}_P)$ and $\text{diag}(\boldsymbol{w}_Q)$) for the sake of clarity. Note in addition that in most examples of this work, the weights are equal to one (B-Spline instead of NURBS) making these matrices equal to the identity.

Finally, let us notice that the refinement matrix does not change as long as the refinement levels of both optimization and analysis models remain unchanged. Thus, the matrix is commonly built once and for all at the beginning of the optimization, and can be reused during the optimization process. In addition, this matrix is sparse and of moderate size (in comparison with the stiffness matrix for instance). As a result, the transition between both models is computationally cheap.

Algorithm 1: Main steps for the analytical discrete sensitivities in IGA-based shape optimization.

```
Inputs : Vector of design variables $\boldsymbol{x}$
       : Definition of the response function $f$
Output: Total derivatives
begin Shape update ..................................................
  | ▸ Shape parametrization                         $\boldsymbol{P} := \boldsymbol{P}(\boldsymbol{x})$                          (22)
  | ▸ Design to analysis model                      $\boldsymbol{Q} = \mathbf{R}\boldsymbol{P}$                             (26)
end
begin Analysis ......................................................
  | ▸ Structural analysis                           $\mathbf{K}\boldsymbol{u} = \boldsymbol{F}$                             (3)
  | ▸ Adjoint analysis                              $\mathbf{K}\boldsymbol{u}^* = \partial f/\partial\boldsymbol{u}$                       (15)
end
begin Gradient calculation ..........................................
  | ▸ Differentiate adjoint work w.r.t. the control points   $\partial \mathrm{W}/\partial\boldsymbol{Q}$          (35)
end
begin Sensitivity propagation .......................................
  | ▸ Analysis to design model                      $\partial \mathrm{W}/\partial\boldsymbol{P} = \mathbf{R}^T\partial \mathrm{W}/\partial\boldsymbol{Q}$                 (28)-(31)
  | ▸ Design model to design variables              $df/dx_i = \partial f/\partial x_i + \partial\boldsymbol{P}/\partial x_i : \partial \mathrm{W}/\partial\boldsymbol{P}$     (24)-(25)
end
return $df/d\boldsymbol{x}$
```

By substituting Eqs. (31) and (24) into the expression of the total derivative (16), it yields the following result:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial \boldsymbol{P}}{\partial x_i} : \mathbf{R}^T \left(\boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} - \boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u} \right). \tag{32}$$

In order to describe the above term in brackets, let us introduce a function W that takes as input arguments the state variables $\boldsymbol{u}$, the adjoint variables $\boldsymbol{u}^*$, and the control points $\boldsymbol{Q}$ of the analysis model. We define this function as:

$$\begin{aligned} \mathrm{W}(\boldsymbol{u}, \boldsymbol{u}^*, \boldsymbol{Q}) &= \mathrm{W}_{\mathrm{ext}}(\boldsymbol{u}^*, \boldsymbol{Q}) + \mathrm{W}_{\mathrm{int}}(\boldsymbol{u}, \boldsymbol{u}^*, \boldsymbol{Q}) \\ &= \boldsymbol{u}^* \cdot \boldsymbol{F}(\boldsymbol{Q}) - \boldsymbol{u}^* \cdot \mathbf{K}(\boldsymbol{Q})\boldsymbol{u}. \end{aligned} \tag{33}$$

Even if we have not yet given the expressions of the stiffness matrix and of the load vector, let us recall that the governing equations (3) come from the virtual work principle. Consequently, function W can be seen as the total virtual work where the virtual displacement field is, in this case, the adjoint solution. Thus, in what follows function W is referred to as the adjoint work.

The load vector and the stiffness matrix depend on the computational domain, and hence, on the control point coordinates since they act on the shape of the domain. Finally, the full computation of the sensitivities requires the partial derivative of the adjoint work W *w.r.t.* the control point coordinates of the analysis model. With these notations in hand, Eq. (32) reads as:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial \boldsymbol{P}}{\partial x_i} : \mathbf{R}^T \frac{\partial \mathrm{W}}{\partial \boldsymbol{Q}}, \tag{34}$$

with

$$\frac{\partial \mathrm{W}}{\partial \boldsymbol{Q}} = \boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} - \boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u}. \tag{35}$$

We will see in the next section how we achieve to compute this missing derivative $\partial \mathrm{W}/\partial \boldsymbol{Q}$. Depending on the formulation of the response function f, the partial derivative $\partial f/\partial x_i$ may also be changed using the applied chain rule of differentiation. In this case the sensitivity reads as:

$$\frac{df}{dx_i} = \frac{\partial \boldsymbol{P}}{\partial x_i} : \mathbf{R}^T \left(\frac{\partial f}{\partial \boldsymbol{Q}} + \frac{\partial \mathrm{W}}{\partial \boldsymbol{Q}} \right). \tag{36}$$

For clarity and due to the large number of equations, we summarize in Algorithm 1 the main steps for the computation of the sensitivity. Once again, all these steps are illustrated in Fig. 1.

4 Differentiating the Element Formulation

This section deals with the computation of the derivatives $\partial \mathrm{W}/\partial \boldsymbol{Q}$ through Eq. (35). These derivatives depend on the element formulation. We present the case of the isogeometric standard solid element and the case of an isogeometric Kirchhoff–Love shell element.

4.1 Element Formulations Using Local Coordinates

A convenient way to compute the derivatives involved in Eq. (35) is to start by formulating the element using the curvilinear formalism. We will see that it enables to better identify the geometric quantities involved in the element formulation in comparison with classical formulation based on Cartesian coordinates. Also, this formalism is applicable, and even required, for a lot of element formulation including beams, shells and solids. Thus, we provide an unified framework usable for a large panel of element formulation. Very similar calculation steps are involved for the two types of isogeometric element considered in this work: the standard solid element (2D or 3D), and a Kirchhoff–Love shell formulation.

Within this context, the position vector associated to a material point is constructed from curvilinear coordinates θ_i instead of Cartesian coordinates X_i. This formalism enables to describe complex shapes with curvatures, and it is therefore widely used in shell theory[5]. Cylindrical and spherical coordinate systems are typical examples of curvilinear coordinate systems. For instance, locating a point belonging to a cylinder is simplified with the use of cylindrical coordinates, i.e. a radial coordinate r, an angular coordinate θ, and a height z:

$$\boldsymbol{X}(r, \theta, z) = r\cos\theta \boldsymbol{E}_1 + r\sin\theta \boldsymbol{E}_2 + z\boldsymbol{E}_3, \tag{37}$$

where $\boldsymbol{X}$ is the vector position associated to the material point, and $\boldsymbol{E}_i$ are Cartesian base vectors. Curvilinear coordinates can finally be seen as a generalization of these types of geometric transformations.

In this context, the gradient operator is written as[5]:

$$\mathrm{Grad}(\mathbf{u}) = \frac{\partial \mathbf{u}}{\partial X_i} \otimes \boldsymbol{E}_i = \frac{\partial \mathbf{u}}{\partial \theta_k} \otimes \frac{\partial \theta_k}{\partial X_i} \boldsymbol{E}_i = \mathbf{u}_{,k} \otimes \boldsymbol{G}^k \tag{38}$$

where the comma subscript represents the partial derivatives *w.r.t.* the curvilinear coordinates, ⊗ denotes the tensor product, and $\boldsymbol{G}^k$ are the contravariant basis vectors associated to the reference configuration. Einstein's summation convention applies here.

4.2 Solid Formulation

4.2.1 Continuum Formulation

The linearized Green-Lagrange strain tensor $\boldsymbol{\varepsilon}$ in curvilinear coordinates reads as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{ij}\, \boldsymbol{G}^i \otimes \boldsymbol{G}^j, \tag{39}$$

where the covariant components $\boldsymbol{\varepsilon}_{ij}$ are given by:

$$\boldsymbol{\varepsilon}_{ij} = \frac{1}{2}\left(\mathbf{u},_i \cdot \boldsymbol{G}_j + \mathbf{u},_j \cdot \boldsymbol{G}_i\right). \tag{40}$$

The covariant basis vectors $\boldsymbol{G}_i$ are tangent to the coordinate lines and are defined as:

$$\boldsymbol{G}_i = \frac{\partial \boldsymbol{X}}{\partial \theta_i} = \boldsymbol{X},_i. \tag{41}$$

The contravariant basis vectors can be expressed in function of the covariant basis vectors as follows:

$$\boldsymbol{G}^i = G^{ij}\boldsymbol{G}_j, \tag{42}$$

where G^{ij} are the contravariant metric coefficients. These coefficients are commonly obtained by the covariant coefficient matrix:

$$\left[\, G^{ij} \,\right] = \left[\, G_{ij} \,\right]^{-1}. \tag{43}$$

Finally, let us give a definition of the covariant metric coefficients G_{ij}. They are computed by the scalar product of covariant basis vectors:

$$G_{ij} = \boldsymbol{G}_i \cdot \boldsymbol{G}_j. \tag{44}$$

More information on these geometric quantities can be found in Echter[25] and Kiendl[48], and in the underlying references cited therein.

There is already an interesting point to notice in the expression of the covariant strain components (40). We can easily identify where are located some information regarding the geometry: all the geometric quantities are contained into the covariant basis vectors only. When using Cartesian coordinates, the linearized strain components would read as:

$$\hat{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{d\mathbf{u}}{dX_j} \cdot \boldsymbol{E}_i + \frac{d\mathbf{u}}{dX_i} \cdot \boldsymbol{E}_j\right). \tag{45}$$

Here the identification of the geometric information is trickier: they are contained into the differential operators. Without giving further details at the moment, it is as of now possible to sense that using the curvilinear formalism will ease later on the differentiation of the element formulation *w.r.t.* the control point coordinates.

Let us resume the element formulation. In the case of small displacements, the second Piola–Kirchhoff stress tensor is approximated by the linearized Cauchy stress tensor $\boldsymbol{\sigma}$. Furthermore, under the assumptions of a linear elastic behavior of the material, the material law is given by Hooke's law:

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}, \tag{46}$$

where $:$ denotes the scalar product of second-order tensors (same convention than in[5]). The fourth-order elasticity tensor is given by:

$$\begin{aligned} \mathbf{C} &= C^{ijkl}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \otimes \boldsymbol{G}_k \otimes \boldsymbol{G}_l, \\ C^{ijkl} &= \lambda G^{ij}G^{kl} + \mu\left(G^{ik}G^{jl} + G^{il}G^{jk}\right). \end{aligned} \tag{47}$$

The material parameters λ and μ are the Lamé constants. Thus, the stress tensor as defined by (46) is usually represented through its contravariant components:

$$\begin{aligned} \boldsymbol{\sigma} &= \boldsymbol{\sigma}^{ij}\boldsymbol{G}_i \otimes \boldsymbol{G}_j\,, \\ \boldsymbol{\sigma}^{ij} &= C^{ijkl}\boldsymbol{\varepsilon}_{kl}\,. \end{aligned} \tag{48}$$

with Einstein's summation convention.

The internal virtual work reads as:

$$\delta \mathrm{W}^{\mathrm{Solid}}_{\mathrm{int}} = -\int_{\bar{\Omega}} \left(\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\right) |J| \mathrm{d}\bar{\Omega}, \tag{49}$$

where the integration variables are directly the curvilinear coordinates θ_i (i.e. $\mathrm{d}\bar{\Omega} = \mathrm{d}\theta_1\mathrm{d}\theta_2\mathrm{d}\theta_3$). The Jacobian $|J|$ can be computed through the following triple product:

$$|J| = \left(\boldsymbol{G}_1 \times \boldsymbol{G}_2\right) \cdot \boldsymbol{G}_3. \tag{50}$$

In addition to the strain components, the curvilinear formalism leads to the expression of the internal virtual work (49) where identifying the geometric quantities is quite straightforward. For comparison, the reader can consult[1] or[37] to observe what are the different transformation steps required to separate the geometric part in the internal work when Cartesian coordinates are used. The control point coordinates act on these geometric quantities. Having a compact expression, as given by the curvilinear formalism, of these quantities will surely simplify the calculation of the requested derivatives $\partial \mathrm{W}/\partial \boldsymbol{Q}$.

Finally, let us give the expression of the external virtual work:

$$\begin{aligned} \delta \mathrm{W}^{\mathrm{Solid}}_{\mathrm{ext}} &= \int_{\Omega} \delta\mathbf{u} \cdot \boldsymbol{F}\mathrm{d}\Omega + \int_{\Gamma} \delta\mathbf{u} \cdot \boldsymbol{t}\mathrm{d}\Gamma \\ &= \int_{\bar{\Omega}} \delta\mathbf{u} \cdot \boldsymbol{F}\,|J|\mathrm{d}\bar{\Omega} + \int_{\bar{\Gamma}} \delta\mathbf{u} \cdot \boldsymbol{t}\,|I|\mathrm{d}\bar{\Gamma}. \end{aligned} \tag{51}$$

where $\boldsymbol{t}$ are the external surface forces, and $\boldsymbol{F}$ are the body forces.

4.2.2 Isogeometric Solid Element

The isogeometric framework sticks well with the use of curvilinear coordinates. One can define those curvilinear coordinates as being the geometric parameters of the spline. The position vector $\boldsymbol{X}$ is defined by a tri-variate mapping:

$$\boldsymbol{X}(\theta_1, \theta_2, \theta_3) = \sum_{k=1}^{m_{cp}} R_k(\theta_1, \theta_2, \theta_3)\boldsymbol{Q}_k. \tag{52}$$

The covariant basis vectors are then obtained by partial derivation with respect to the parameters θ_i which gives:

$$\boldsymbol{G}_i(\theta_1, \theta_2, \theta_3) = \sum_{k=1}^{m_{cp}} R_{k,i}(\theta_1, \theta_2, \theta_3)\boldsymbol{Q}_k. \tag{53}$$

With these vectors in hand, one can compute the covariant metric coefficients G_{ij} using Eq. (44). Then, the contravariant metric coefficients G^{ij} are obtained by inverting a 3-by-3 matrix as given by Eq. (43). Those are the main calculation steps of the formulation. Hence, using curvilinear coordinates is actually not a complex task, especially with IGA.

For the rest, everything is identical to classical Finite Element formulations. The displacement field is approximated using the basis functions coming from the discretization of the geometry:

$$\mathbf{u}^h(\theta_1, \theta_2, \theta_3) = \sum_{k=1}^{m_{cp}} R_k(\theta_1, \theta_2, \theta_3)\boldsymbol{u}_k. \tag{54}$$

It leads to the following expression of the discretized strain components:

$$\boldsymbol{\varepsilon}_{ij}^h = \sum_{k=1}^{m_{cp}} \frac{1}{2}\left(R_{k,i}\boldsymbol{G}_j + R_{k,j}\boldsymbol{G}_i\right) \cdot \boldsymbol{u}_k, \tag{55}$$

which can be stored into a vector by using, for example, the Voigt notation:

$$\boldsymbol{\varepsilon}^h = \sum_{k=1}^{m_{cp}} \mathbf{B}_k\boldsymbol{u}_k, \tag{56}$$

where the $\mathbf{B}_k$ are strain matrices that contain the terms indicated in brackets in Eq. (55). Identically, the discretized stress is given by:

$$\boldsymbol{\sigma}^h = \sum_{k=1}^{m_{cp}} \mathbf{H}\mathbf{B}_k\boldsymbol{u}_k, \tag{57}$$

where $\mathbf{H}$ is the matrix representation of the fourth-order material tensor given in Eq. (47).

Finally, introducing these approximated quantities into the virtual works (49) and (51) lead to the so-called stiffness matrix $\mathbf{K}$ and load vector $\boldsymbol{F}$ which can be obtained by assembling elementary matrices of the form:

$$\mathbf{K}_{kl}^e = \int_{\bar{\Omega}^e} \mathbf{B}_k{}^T\mathbf{H}\mathbf{B}_l|J|\mathrm{d}\bar{\Omega}, \tag{58}$$

$$\boldsymbol{F}_k^e = \int_{\bar{\Omega}^e} R_k\,\boldsymbol{F}\,|J|\mathrm{d}\bar{\Omega} + \int_{\bar{\Gamma}^e} R_k\,\boldsymbol{t}\,|I|\mathrm{d}\bar{\Gamma}. \tag{59}$$

The integral is computed numerically using quadrature rule. Even if the intermediary steps are a bit different, one should keep in mind that the solid element written in the curvilinear fashion is strictly the same than the standard solid element. At the end of the day, the stiffness matrices obtained with both approaches are identical. However, we will see that using the curvilinear formalism provides a suitable way to compute analytically the derivatives involved in the sensitivities [see Eq. (35)].

4.2.3 Differentiating the Standard IGA Operators

According to Eq. (35), we are at a point where we need to compute the derivatives of the external and internal works:

$$\frac{\partial \mathrm{W}_{\mathrm{ext}}}{\partial \boldsymbol{Q}} = \boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} \quad \text{and} \quad \frac{\partial \mathrm{W}_{\mathrm{int}}}{\partial \boldsymbol{Q}} = -\boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}}\boldsymbol{u}. \tag{60}$$

The stiffness matrix and the load vector are built elementwise. The sensitivity analysis will be performed similarly, and we focus here mainly on the stiffness matrix because it is the most challenging part. Summing over each element gives:

$$\boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}}\boldsymbol{u} = \sum_e \left(\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}}\boldsymbol{u}^e\right). \tag{61}$$

An important point to notice is that only the control points associated to the current element e give non-zero derivatives in the term $\partial\mathbf{K}^e/\partial\boldsymbol{Q}$. Thus, for each element of the analysis model, only the few corresponding components of the gradient are updated. Moreover, for each element, it is numerically not efficient to build a large matrix $\partial\mathbf{K}^e/\partial\boldsymbol{Q}$. Instead, we use the following development:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}}\boldsymbol{u}^e = \sum_k \sum_l \boldsymbol{u}_k^{e*} \cdot \frac{\partial \mathbf{K}_{kl}^e}{\partial \boldsymbol{Q}}\boldsymbol{u}_l^e, \tag{62}$$

where k and l denote two control point indices associated to the current element e. Using Eq. (58), the derivatives of the components of the elementary stiffness matrices *w.r.t.* the control points are given by:

$$\frac{\partial \mathbf{K}^e_{kl}}{\partial \boldsymbol{Q}} = \int_{\bar{\Omega}^e} \left(\frac{\partial {\mathbf{B}_k}^T}{\partial \boldsymbol{Q}} \mathbf{H}\mathbf{B}_l + {\mathbf{B}_k}^T \frac{\partial \mathbf{H}}{\partial \boldsymbol{Q}} \mathbf{B}_l + {\mathbf{B}_k}^T \mathbf{H} \frac{\partial \mathbf{B}_l}{\partial \boldsymbol{Q}} \right) |J| \mathrm{d}\bar{\Omega} + \int_{\bar{\Omega}^e} {\mathbf{B}_k}^T \mathbf{H}\mathbf{B}_l \frac{\partial |J|}{\partial \boldsymbol{Q}} \mathrm{d}\bar{\Omega}. \tag{63}$$

We can put this last equation into Eq. (62). By commuting the double sum and the integral, we get:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}^e = \int_{\bar{\Omega}^e} \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}} : \boldsymbol{\sigma} + \boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}} \right) |J| \mathrm{d}\bar{\Omega} + \int_{\bar{\Omega}^e} (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \mathrm{d}\bar{\Omega}. \tag{64}$$

Three terms can be identified:

1. *Derivative of the Jacobian*

$$(\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \tag{65}$$

where

$$\boldsymbol{\varepsilon}^* = \sum_k \mathbf{B}_k \boldsymbol{u}^*_k, \quad \boldsymbol{\sigma} = \sum_l \mathbf{H}\mathbf{B}_l \boldsymbol{u}_l, \tag{66}$$

2. *Derivative of the (adjoint) Strains*

$$\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}} = \sum_k \frac{\partial \mathbf{B}_k}{\partial \boldsymbol{Q}} \boldsymbol{u}^*_k, \tag{67}$$

3. *Derivative of the Stresses*

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}} = \sum_l \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{Q}} \mathbf{B}_l + \mathbf{H} \frac{\partial \mathbf{B}_l}{\partial \boldsymbol{Q}} \right) \boldsymbol{u}_l. \tag{68}$$

We finally end up with a generic expression of the analytical sensitivities. However, to the author's understanding, the partial derivatives *w.r.t.* the control points are more accessible in the context of element formulation based on curvilinear coordinates as shown in what follows.

Derivative of the Jacobian Differentiating the Jacobian *w.r.t.* the control points leads to:

$$\frac{\partial |J|}{\partial \boldsymbol{Q}} = (\boldsymbol{G}_2 \times \boldsymbol{G}_3) \cdot \frac{\partial \boldsymbol{G}_1}{\partial \boldsymbol{Q}} + (\boldsymbol{G}_3 \times \boldsymbol{G}_1) \cdot \frac{\partial \boldsymbol{G}_2}{\partial \boldsymbol{Q}} + (\boldsymbol{G}_1 \times \boldsymbol{G}_2) \cdot \frac{\partial \boldsymbol{G}_3}{\partial \boldsymbol{Q}}. \tag{69}$$

We remind that the partial derivative $\partial / \partial \boldsymbol{Q}$ contains the partial derivatives *w.r.t.* the three components of the all active control points (denoted previously Q_{ja} which corresponds to the jth component of the control point number a). Differentiating the covariant basis vectors $\boldsymbol{G}_i$ gives:

$$\frac{\partial \boldsymbol{G}_i}{\partial Q_{ja}} = R_{a,i}\, \boldsymbol{e}_j, \tag{70}$$

and finally, the complete expression for the derivatives of the Jacobian is:

$$\frac{\partial |J|}{\partial \boldsymbol{Q}_a} = R_{a,1} \boldsymbol{G}_2 \times \boldsymbol{G}_3 + R_{a,2} \boldsymbol{G}_3 \times \boldsymbol{G}_1 + R_{a,3} \boldsymbol{G}_1 \times \boldsymbol{G}_2. \tag{71}$$

The derivative $\partial |J| / \partial \boldsymbol{Q}_a$ takes the form of a vector with three components. The gradient $\partial |J| / \partial \boldsymbol{Q}$ in Eq. (64) collects these derivatives where the index a corresponds to the active control points (those from the current element e).

Derivative of the Strains It is computationally not efficient to build the matrices $\partial \mathbf{B}_k / \partial \boldsymbol{Q}$ and then summing as expressed in Eq. (67). If one takes a closer look at the expression, one can see that it is possible to commute the derivative and the sum. For instance, let us take the case of a specific strain component $\boldsymbol{\varepsilon}^h_{ij}$:

$$\frac{\partial \boldsymbol{\varepsilon}^h_{ij}}{\partial \boldsymbol{Q}} = \sum_{k=1}^{m_{cp}} \frac{1}{2} \left(R_{k,i} \frac{\partial \boldsymbol{G}_j}{\partial \boldsymbol{Q}} + R_{k,j} \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{Q}} \right) \cdot \boldsymbol{u}_k = \frac{1}{2} \left({\mathbf{u}^h}_{,i} \cdot \frac{\partial \boldsymbol{G}_j}{\partial \boldsymbol{Q}} + \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{Q}} \cdot {\mathbf{u}^h}_{,j} \right). \tag{72}$$

We already know the derivative of the covariant vectors *w.r.t.* the control points, e.g. see Eq. (70). Finally the term in Eq. (64) with the derivative of the (adjoint) strains is not as hard as it may seem. One has to compute quantities of the following form:

$$\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}_a} : \boldsymbol{\sigma} = \sum_{ij} \frac{1}{2} \sigma^{ij} \left({\mathbf{u}^*}_{,i} R_{a,j} + R_{a,i} {\mathbf{u}^*}_{,j} \right), \tag{73}$$

where the index a corresponds to the active control points, and ${\mathbf{u}^*}_{,i}$ reads as:

$${\mathbf{u}^*}_{,i} = \sum_{k=1}^{m_{cp}} R_{k,i} \boldsymbol{u}^*_k\,. \tag{74}$$

The result given by Eq. (73) should be seen as a vector with three components (derivation *w.r.t.* each of the three components of $\boldsymbol{Q}_a$). We end up with a compact expression of the term involving the derivatives of the strain components. One simply has to compute the derivatives of the adjoint field as expressed in Eq. (74), and then put these results in Eq. (73). In Qian[75] where Cartesian coordinates are used, such a compact expression is not provided. Instead, several intermediary results are given since combining them leads to a long and non-practicable expression.

Derivative of the Stresses Having the derivatives of the strains in hand is already a first step to compute those of the stresses. Again, Eq. (68) is not used as such. Instead, we commute the derivative and the sum. For a given stress component, the differentiation *w.r.t.* the control points gives:

$$\frac{\partial \sigma^{ij}}{\partial \boldsymbol{Q}} = \frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}} \boldsymbol{\varepsilon}_{kl} + C^{ijkl} \frac{\partial \boldsymbol{\varepsilon}_{kl}}{\partial \boldsymbol{Q}}, \tag{75}$$

with Einstein's summation convention.

The derivatives of the material tensor $\partial C^{ijkl} / \partial \boldsymbol{Q}$ are obtained by the derivation of the Eq. (47). A chain rule of differentiation leads to:

$$\begin{aligned} \frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}} = \lambda \left(\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} G^{kl} + G^{ij} \frac{\partial G^{kl}}{\partial \boldsymbol{Q}} \right) \\ + \mu \left(\frac{\partial G^{ik}}{\partial \boldsymbol{Q}} G^{jl} + G^{ik} \frac{\partial G^{jl}}{\partial \boldsymbol{Q}} + \frac{\partial G^{il}}{\partial \boldsymbol{Q}} G^{jk} + G^{il} \frac{\partial G^{jk}}{\partial \boldsymbol{Q}} \right). \end{aligned} \tag{76}$$

Thus, the derivatives of the contravariant metrics *w.r.t.* to the control points are involved. We have not yet computed these derivatives. To this purpose, let us remind that these metrics are obtained by inverting a 3-by-3 matrix which takes the covariant metrics as components [see Eq. (43)]. One can notice that the following relation holds true:

$$\left[G^{ij} \right] \left[G_{ij} \right] = \mathbf{I}, \tag{77}$$

where **I** denotes here the identity matrix of size 3. Differentiating this last equation leads to:

$$\left[\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} \right] \left[G_{ij} \right] + \left[G^{ij} \right] \left[\frac{\partial G_{ij}}{\partial \boldsymbol{Q}} \right] = \left[0 \right]. \tag{78}$$

Then, we multiply this result on the right with the matrix containing the contravariant metrics in order to identify the derivatives of the contravariant metrics. Using Eq. (77) finally yields:

$$\left[\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} \right] = - \left[G^{ij} \right] \left[\frac{\partial G_{ij}}{\partial \boldsymbol{Q}} \right] \left[G^{ij} \right]. \tag{79}$$

This enables to express the derivatives of the contravariant metrics as functions of the derivatives of the covariant metrics:

$$\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} = - \sum_{kl} G^{ik} G^{lj} \frac{\partial G_{kl}}{\partial \boldsymbol{Q}}. \tag{80}$$

The covariant metrics are obtained by dot products between the covariant vectors as given in Eq. (44). Thus, the derivatives of the covariant metrics *w.r.t.* the control points is quite straightforward:

$$\frac{\partial G_{kl}}{\partial \boldsymbol{Q}} = \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{Q}} \cdot \boldsymbol{G}_l + \boldsymbol{G}_k \cdot \frac{\partial \boldsymbol{G}_l}{\partial \boldsymbol{Q}}, \tag{81}$$

and by introducing Eq. (70) one can get the results we were interested in:

$$\frac{\partial G_{kl}}{\partial \boldsymbol{Q}_a} = R_{a,k} \boldsymbol{G}_l + R_{a,l} \boldsymbol{G}_k. \tag{82}$$

We now have all the ingredients to compute the derivatives of the stresses *w.r.t.* the control points of the analysis model; i.e. we know how to compute the terms involved in Eq. (75). As for the strain components, we end up with a compact expression of the derivatives of stress components thanks to the use of the curvilinear formalism. It involves the derivatives of the contravariant metrics contained in the components of the material tensor. The equations to be used are (76), (80) and (82).

Algorithm 2: Computation of the partial derivatives of the total work *w.r.t.* the control point coordinates $\partial \mathrm{W}/\partial \boldsymbol{Q}$.

```
Input  : State and Adjoint variables $\boldsymbol{u}$ and $\boldsymbol{u}^*$
       : Geometric and Mechanical properties
Output: Gradient
gradW(:, :) = 0. // Initialize with size $(m_{cp} \times 3)$
for $i_e = 1$ to $n_e$ do
    gradWe(:, :) = 0. // Initialize with size $(m^e_{cp} \times 3)$
    for $i_{gp} = 1$ to $n_{gp}$ do

        Evaluate basis function and derivatives at gauss point $i_{gp}$
        Compute curvilinear quantities $(\boldsymbol{G}_i, \boldsymbol{G}^i, G_{ij}, G^{ij})$
        Compute derivatives of displacement and adjoint fields $(\mathbf{u}_{,i}$ and $\mathbf{u}^*{}_{,i})$
        Infer state and adjoint strains and stresses $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}, \boldsymbol{\varepsilon}^*, \boldsymbol{\sigma}^*)$
        Compute and store other redundant quantities $(|J|, (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma})$, etc.)

        for $a = 1$ to $m^e_{cp}$ do
            Compute derivative of Jacobian ........................ Equation (71):
                $\frac{\partial |J|}{\partial \boldsymbol{Q}_a} = R_{a,1}\boldsymbol{G}_2 \times \boldsymbol{G}_3 \; + \; R_{a,2}\boldsymbol{G}_3 \times \boldsymbol{G}_1 \; + \; R_{a,3}\boldsymbol{G}_1 \times \boldsymbol{G}_2$
            Compute derivative of state and adjoint strains ........................ Equation (73):
                $\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}_a} : \boldsymbol{\sigma} = \sum_{ij} \frac{1}{2}\sigma^{ij}\left(\mathbf{u}^*{}_{,i}\, R_{a,j} \; + \; R_{a,i}\, \mathbf{u}^*{}_{,j}\right)$
                $\boldsymbol{\sigma}^* : \frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{Q}_a} = \sum_{ij} \frac{1}{2}\sigma^{*ij}\left(\mathbf{u}_{,i}\, R_{a,j} \; + \; R_{a,i}\, \mathbf{u}_{,j}\right)$
            Compute derivative of material tensor ........................ Equations (76),(80),(82):
                $\frac{\partial G^{ij}}{\partial \boldsymbol{Q}_a} = -\sum_{kl} G^{ik}G^{lj}\left(R_{a,k}\boldsymbol{G}_l + R_{a,l}\boldsymbol{G}_k\right)$
                $\frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}_a} = \lambda\left(\frac{\partial G^{ij}}{\partial \boldsymbol{Q}_a}G^{kl} + G^{ij}\frac{\partial G^{kl}}{\partial \boldsymbol{Q}_a}\right) + \mu\left(\frac{\partial G^{ik}}{\partial \boldsymbol{Q}_a}G^{jl} + G^{ik}\frac{\partial G^{jl}}{\partial \boldsymbol{Q}_a} + \frac{\partial G^{il}}{\partial \boldsymbol{Q}_a}G^{jk} + G^{il}\frac{\partial G^{jk}}{\partial \boldsymbol{Q}_a}\right)$
            Compute derivative of stresses ........................ Equation (75):
                $\boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}_a} = \boldsymbol{\sigma}^* : \frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{Q}_a} \; + \; \sum_{ij}\sum_{kl}\left(\boldsymbol{\varepsilon}^*_{ij}\frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}_a}\boldsymbol{\varepsilon}_{kl}\right)$
            Update 3 components of the gradient associated to $\boldsymbol{Q}_a$ (i.e. add terms to gradWe$(a, :)$) ... Equation (84):
                $\frac{\partial \mathrm{W}^e}{\partial \boldsymbol{Q}_a} \mathrel{-}= \left\{\left(\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}\right)\frac{\partial |J|}{\partial \boldsymbol{Q}_a} + \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}_a} : \boldsymbol{\sigma} + \boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}_a}\right)|J|\right\}|J_{gp}|w_{gp}$
            If exist, add terms coming from the body forces ........................ Equation (85):
                $\frac{\partial \mathrm{W}^e}{\partial \boldsymbol{Q}_a} \mathrel{+}= \left\{\left(\mathbf{u}^* \cdot \boldsymbol{f}\right)\frac{\partial |J|}{\partial \boldsymbol{Q}_a} + \left(\mathbf{u}^* : \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{Q}_a}\right)|J|\right\}|J_{gp}|w_{gp}$
        end
    end
    gradW(activeCPs, :) += gradWe(:, :)
end
If exist, add terms coming from external surface forces (same strategy than for the body forces) ..... Equation (86).
return gradW
```

4.2.4 Summary and Generalization

Despite the large number of equations, we end up with a general expression of the required derivatives (formulated initially by Eq. (35)). It reads as the differentiation of the total work evaluated using the displacement and the adjoint fields *w.r.t.* the control point coordinates:

$$\frac{\partial \mathrm{W}}{\partial \boldsymbol{Q}} = \frac{\partial \mathrm{W}_{\text{int}}}{\partial \boldsymbol{Q}}\left(\mathbf{u}^{*h}, \mathbf{u}^{h}\right) + \frac{\partial \mathrm{W}_{\text{ext}}}{\partial \boldsymbol{Q}}\left(\mathbf{u}^{*h}\right), \tag{83}$$

where the derivatives of the internal work are given by:

$$\begin{aligned}\frac{\partial \mathrm{W}_{\text{int}}}{\partial \boldsymbol{Q}} = &- \int_{\bar{\Omega}} \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}} : \boldsymbol{\sigma} + \boldsymbol{\varepsilon} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}} \right) |J| \mathrm{d}\bar{\Omega} \\ &- \int_{\bar{\Omega}} (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \mathrm{d}\bar{\Omega},\end{aligned} \tag{84}$$

and the derivatives of the external work are given by:

$$\frac{\partial \mathrm{W}_{\text{ext}}}{\partial \boldsymbol{Q}} = \int_{\bar{\Omega}} \left\{ \mathbf{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} |J| + (\mathbf{u}^* \cdot \boldsymbol{F}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega} \tag{85}$$

$$+ \int_{\bar{\Gamma}} \left\{ \mathbf{u}^* \cdot \frac{\partial \boldsymbol{t}}{\partial \boldsymbol{Q}} |I| + (\mathbf{u}^* \cdot \boldsymbol{t}) \frac{\partial |I|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Gamma}. \tag{86}$$

Regarding the implementation, one should view the derivatives as given by these equations. We present in Algorithm 2 the main steps for the implementation of the aforementioned derivatives. We summarize the principal equations one would required to implement the gradient.

4.3 Shell Formulation

4.3.1 Continuum Formulation

The Kirchhoff–Love formulation has been largely studied in the literature both for analysis and for shape optimization. We only remind here key theoretical points in order to be able to present the analytical sensitivities. The starting point consists in invoking specific kinematic assumptions, namely the Kirchhoff–Love hypotheses. These hypotheses introduced by Kirchhoff[51] and Love[61] state that the normals to the mid-surface in the reference configuration remain normal and unstretched in the deformed configuration.

There are different strategies to impose the kinematic assumptions. In this work we follow the strategy from Kiendl et al.[49] based on the direct approach. It means that the shell is regarded from the beginning as a two-dimensional surface (often named as a Cosserat surface) and proper kinematic assumptions, representing the three-dimensional behavior, are postulated. Thus, the shell continuum is described by its mid-surface $\boldsymbol{S}$ and director vectors:

$$\boldsymbol{X}(\theta_1, \theta_2, \theta_3) = \boldsymbol{S}(\theta_1, \theta_2) + \theta_3 \boldsymbol{A}_3(\theta_1, \theta_2), \;\; \theta_3 \in [\text{-}\tfrac{t}{2}, \tfrac{t}{2}], \tag{87}$$

where t is the shell thickness. In the case of Kirchhoff–Love kinematic, the director vector is taken as the normal at each point of the mid-surface:

$$\boldsymbol{A}_3 = \frac{1}{A} \boldsymbol{A}_1 \times \boldsymbol{A}_2, \quad \text{with} \quad A = |\boldsymbol{A}_1 \times \boldsymbol{A}_2|. \tag{88}$$

where $\boldsymbol{A}_\alpha$, $\alpha = 1, 2$ are the covariant vectors associated to the mid-surface.

By introducing the aforementioned kinematic assumptions, the displacement field of the entire body can be described only by the displacement $\mathbf{u}$ of the mid-surface:

$$\mathbf{u}^{3D}(\theta_1, \theta_2, \theta_3) = \mathbf{u}(\theta_1, \theta_2) + \theta_3 \boldsymbol{w}^{\text{lin}}(\theta_1, \theta_2). \tag{89}$$

In this work we use the linearized difference vector $\boldsymbol{w}^{\text{lin}}$ which is valid under the assumption of small displacement[26, 49]. This vector is also expressed *w.r.t.* the displacement of the mid-surface. The kinematic assumptions also involve that transversal strains vanish (trough the thickness). More details can be found, for example in Kiendl[48].

Let us directly give the expression of the virtual works for the Kirchhoff–Love shell formulation. It reads as:

$$\delta \mathrm{W}_{\text{int}}^{\text{KL}} = - \int_{\bar{\Omega}_0} \left(\boldsymbol{n} : \delta \boldsymbol{e} + \boldsymbol{m} : \delta \boldsymbol{\kappa} \right) A \mathrm{d}\bar{\Omega}_0, \tag{90}$$

$$\delta \mathrm{W}_{\text{ext}}^{\text{KL}} = \int_{\bar{\Omega}_0} \delta \mathbf{u} \cdot \boldsymbol{p} \, A \mathrm{d}\bar{\Omega}_0 + \int_{\Gamma_0} \delta \mathbf{u} \cdot \boldsymbol{t} \, \mathrm{d}\Gamma_0, \tag{91}$$

where A is expressed in (88), $\boldsymbol{p}$ denotes distributed loads per unit of area applied on the mid-surface Ω_0 , and $\boldsymbol{t}$ denotes axial forces per unit of length applied on the edges of the patch Γ_0. The internal work is expressed as the sum of two contributions: the membrane and the bending part. The membrane and bending strains are formulated using local coordinates as we did for the standard formulation in Sect. 4.2. More precisely, the covariant membrane components are:

$$e_{\alpha\beta} = \frac{1}{2} \left(\mathbf{u}_{,\alpha} \cdot \boldsymbol{A}_\beta + \mathbf{u}_{,\beta} \cdot \boldsymbol{A}_\alpha \right). \tag{92}$$

Greek indices (α, β) takes on values 1 or 2. The covariant bending components are:

$$\begin{aligned}\boldsymbol{\kappa}_{\alpha\beta} = &-\mathbf{u}_{,\alpha\beta} \cdot \boldsymbol{A}_3 \\ &+ \frac{1}{A} \Big(\mathbf{u}_{,1} \cdot (\boldsymbol{A}_{\alpha,\beta} \times \boldsymbol{A}_2) + \mathbf{u}_{,2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha,\beta}) \Big) \\ &+ \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha,\beta}}{A} \Big(\mathbf{u}_{,1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \mathbf{u}_{,2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1) \Big).\end{aligned} \tag{93}$$

In the expression of the virtual work (90), $\boldsymbol{n}$ and $\boldsymbol{m}$ denote the normal forces and the bending moments respectively. They are expressed as follows:

$$\boldsymbol{n} = t\,\mathbf{C}_0 : \boldsymbol{e} \quad n^{\alpha\beta} = \frac{tE}{1-\nu^2} H_0^{\alpha\beta\gamma\delta} e_{\gamma\delta}, \tag{94}$$

$$\boldsymbol{m} = \frac{t^3}{12}\mathbf{C}_0 : \boldsymbol{\kappa} \quad m^{\alpha\beta} = \frac{t^3 E}{12(1-\nu^2)} H_0^{\alpha\beta\gamma\delta} \boldsymbol{\kappa}_{\gamma\delta}, \tag{95}$$

with:

$$H_0^{\alpha\beta\gamma\delta} = \nu A^{\alpha\beta}A^{\gamma\delta} + \frac{1}{2}(1-\nu)\left(A^{\alpha\gamma}A^{\beta\delta} + A^{\alpha\delta}A^{\beta\gamma}\right). \tag{96}$$

The constitutive tensor $\mathbf{C}_0$ includes the plane-stress condition through condensation of the material equations[5]. It reads as:

$$C_0^{\alpha\beta\gamma\delta} = \bar{\lambda}A^{\alpha\beta}A^{\gamma\delta} + \mu\left(A^{\alpha\gamma}A^{\beta\delta} + A^{\alpha\delta}A^{\beta\gamma}\right), \tag{97}$$

where $\bar{\lambda} = 2\lambda\mu/(\lambda + 2\mu)$.

4.3.2 Isogeometric Kirchhoff–Love Element

The Kirchhoff–Love NURBS element is obtained by discretizing the mid-surface with a NURBS surface. This discretization is also used to approximate the mid-surface displacement field:

$$\begin{aligned} \boldsymbol{S}(\theta_1,\theta_2) &= \sum_{k=1}^{m_{cp}} R_k(\theta_1,\theta_2)\boldsymbol{Q}_k, \\ \mathbf{u}^h(\theta_1,\theta_2) &= \sum_{k=1}^{m_{cp}} R_k(\theta_1,\theta_2)\boldsymbol{u}_k. \end{aligned} \tag{98}$$

Then, the discretized membrane and bending strains take the following forms:

$$\boldsymbol{e}^h = \sum_{k=1}^{m_{cp}} \mathbf{B}_k^m \boldsymbol{u}_k, \tag{99}$$

$$\boldsymbol{\kappa}^h = \sum_{k=1}^{m_{cp}} \mathbf{B}_k^b \boldsymbol{u}_k. \tag{100}$$

The expression of the membrane strain matrices $\mathbf{B}_k^m$ and of the bending strain matrices $\mathbf{B}_k^b$ can be inferred from Eqs. (92) and (93).

The stiffness matrix of the Kirchhoff–Love shell formulation can be built through 3-by-3 matrices of the form:

$$\mathbf{K}_{kl}^e = \int_{\bar{\Omega}_0^e} \left[t\mathbf{B}_k^{mT}\mathbf{H}_0\mathbf{B}_l^m + \frac{t^3}{12}\mathbf{B}_k^{bT}\mathbf{H}_0\mathbf{B}_l^b \right] A\mathrm{d}\theta_1\mathrm{d}\theta_2, \tag{101}$$

where k and l are indices of two control points related to element e, and the matrix $\mathbf{H}_0$ reads as:

$$\mathbf{H}_0 = \frac{E}{1-\nu^2}\begin{bmatrix} H_0^{11\,11} & H_0^{11\,22} & H_0^{11\,12} \\ * & H_0^{22\,22} & H_0^{22\,12} \\ * & * & H_0^{12\,12} \end{bmatrix}, \tag{102}$$

and where the components $H_0^{\alpha\beta\gamma\delta}$ are given by Eq. (96). The integral is later computed using numerical integration. The load vector, which expresses the external virtual work (91) once the displacement field is discretized, reads as:

$$\boldsymbol{F}_k = \int_{\Omega_0} R_k\boldsymbol{p}\,\mathrm{d}\Omega_0 + \int_{\Gamma_0} R_k\boldsymbol{t}\,\mathrm{d}\Gamma_0. \tag{103}$$

4.3.3 Differentiating IGA Kirchhoff–Love Operators

For the Kirchhoff–Love shell formulation, we follow the same logic as for the standard IGA formulation we have just dealt with (see Sect. 4.2). Thus, here we skip redundant calculation steps. Especially, one can obtain the counterpart of Eq. (64) for the Kirchhoff–Love shell formulation by applying the same reasoning. In that respect, we can show that the derivatives of the adjoint internal work *w.r.t.* the control point coordinates are given by:

$$\begin{aligned} \frac{\partial W_{\mathrm{int}}^{\mathrm{KL}}}{\partial \boldsymbol{Q}}\left(\mathbf{u}^{*h},\mathbf{u}^h\right) = &\int_{\bar{\Omega}_0} \left(\frac{\partial \boldsymbol{e}^*}{\partial \boldsymbol{Q}} : \boldsymbol{n} + \boldsymbol{e}^* : \frac{\partial \boldsymbol{n}}{\partial \boldsymbol{Q}}\right) A\mathrm{d}\bar{\Omega}_0 \\ &+ \int_{\bar{\Omega}_0} \left(\frac{\partial \boldsymbol{\kappa}^*}{\partial \boldsymbol{Q}} : \boldsymbol{m} + \boldsymbol{\kappa}^* : \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{Q}}\right) A\mathrm{d}\bar{\Omega}_0 \\ &+ \int_{\bar{\Omega}_0} (\boldsymbol{e}^* : \boldsymbol{n} + \boldsymbol{\kappa}^* : \boldsymbol{m}) \frac{\partial A}{\partial \boldsymbol{Q}} \mathrm{d}\bar{\Omega}_0. \end{aligned} \tag{104}$$

Several terms can be identified: we need to compute the derivatives of the Jacobian, the derivatives of the (adjoint) strains (membrane and bending), and the derivatives of the stress resultants (membrane and bending).

Derivative of the Jacobian We already introduced the expression of the derivatives of the Jacobian *w.r.t.* the control points in case of a volume, see Eq. (71). In case of a surface, the derivatives are given by:

$$\frac{\partial A}{\partial \boldsymbol{Q}_a} = R_{a,1}\boldsymbol{A}_2 \times \boldsymbol{A}_3 + R_{a,2}\boldsymbol{A}_3 \times \boldsymbol{A}_1. \tag{105}$$

One has to differentiate Eq. (88) to get this result.

Derivative of the Membrane Strains and Stresses The expressions of the membrane strains $\boldsymbol{e}_{\alpha\beta}$ and the membrane forces $\boldsymbol{n}^{\alpha\beta}$ involved in the Kirchhoff–Love shell are essentially similar to the strain and stress fields of the standard solid elements (2D problem). Thus, we give here only the final results. One can recover the following equations by going through what has been presented for the solid formulation. For the derivatives of the membrane strains, we obtain:

$$\frac{\partial \boldsymbol{e}^*_{\alpha\beta}}{\partial \boldsymbol{Q}_a} = \frac{1}{2}\mathbf{u}^*{,_\alpha} R_{a,\beta} \, + \, R_{a,\alpha}\mathbf{u}^*{,_\beta} \tag{106}$$

where the adjoint solution is built using the bi-variate basis functions and takes the same form as Eq. (74).

The derivatives of the membrane forces is obtained using the constitutive Eq. (94). We have:

$$\frac{\partial \boldsymbol{n}^{\alpha\beta}}{\partial \boldsymbol{Q}} = t\,\frac{\partial C_0^{\alpha\beta\gamma\delta}}{\partial \boldsymbol{Q}}\boldsymbol{e}_{\gamma\delta} \, + \, t\,C_0^{\alpha\beta\gamma\delta}\frac{\partial \boldsymbol{e}_{\gamma\delta}}{\partial \boldsymbol{Q}}. \tag{107}$$

The derivatives of the material tensor $\partial C_0^{\alpha\beta\gamma\delta}/\partial \boldsymbol{Q}$ can be computed similarly to what has been done for the 3D constitutive law (76).

Since the membrane part involved in the analytical sensitivities for the Kirchhoff–Love formulation is very similar to classical 2D problems (e.g. by taking thickness equals to one and assuming plane-stress state), the interested readers can start by implementing the analytical sensitivities in that context.

Derivative of the Bending Strains and Stresses The bending part involved in the sensitivity (104) requires additional developments. However, the core idea remains the same. A chain rule is applied until we get an expression with quantities that we know how to derive *w.r.t.* the control points. Let us split the expression of the bending strain (93) into three terms as follows in order to describe the derivatives:

$$\boldsymbol{\kappa}_{\alpha\beta} = -\mathbf{u}{,_{\alpha\beta}} \cdot \boldsymbol{A}_3 \, + \, \boldsymbol{\kappa}^1_{\alpha\beta} \, + \, \boldsymbol{\kappa}^2_{\alpha\beta}, \tag{108}$$

where

$$\begin{aligned} \boldsymbol{\kappa}^1_{\alpha\beta} &= \frac{1}{A}\Big(\mathbf{u}{,_1} \cdot (\boldsymbol{A}_{\alpha{,}\beta} \times \boldsymbol{A}_2) + \mathbf{u}{,_2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha{,}\beta})\Big), \\ \boldsymbol{\kappa}^2_{\alpha\beta} &= \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha{,}\beta}}{A}\Big(\mathbf{u}{,_1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \mathbf{u}{,_2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1)\Big). \end{aligned} \tag{109}$$

Thus, the derivatives of the bending strains *w.r.t.* the control points are given by:

$$\frac{\partial \boldsymbol{\kappa}_{\alpha\beta}}{\partial \boldsymbol{Q}} = -\mathbf{u}{,_{\alpha\beta}} \cdot \frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}} \, + \, \frac{\partial \boldsymbol{\kappa}^1_{\alpha\beta}}{\partial \boldsymbol{Q}} \, + \, \frac{\partial \boldsymbol{\kappa}^2_{\alpha\beta}}{\partial \boldsymbol{Q}}. \tag{110}$$

The derivatives $\partial \boldsymbol{\kappa}^1_{\alpha\beta}/\partial \boldsymbol{Q}$ can be written as follows:

$$\begin{aligned} \frac{\partial \boldsymbol{\kappa}^1_{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{\partial A^{-1}}{\partial \boldsymbol{Q}}&\Big(\mathbf{u}{,_1} \cdot (\boldsymbol{A}_{\alpha{,}\beta} \times \boldsymbol{A}_2) + \mathbf{u}{,_2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha{,}\beta})\Big) \\ &+ \frac{1}{A}\Big((\mathbf{u}{,_1} \times \boldsymbol{A}_{\alpha{,}\beta}) \cdot \frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}} - (\mathbf{u}{,_1} \times \boldsymbol{A}_2) \cdot \frac{\partial \boldsymbol{A}_{\alpha{,}\beta}}{\partial \boldsymbol{Q}}\Big) \\ &+ \frac{1}{A}\Big((\boldsymbol{A}_{\alpha{,}\beta} \times \mathbf{u}{,_2}) \cdot \frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}} - (\boldsymbol{A}_1 \times \mathbf{u}{,_2}) \cdot \frac{\partial \boldsymbol{A}_{\alpha{,}\beta}}{\partial \boldsymbol{Q}}\Big), \end{aligned} \tag{111}$$

where some circular shifts have been performed in the scalar triple products. Identically, the other term $\partial \boldsymbol{\kappa}^2_{\alpha\beta}/\partial \boldsymbol{Q}$ is given by:

$$\begin{aligned} \frac{\partial \boldsymbol{\kappa}^2_{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{\partial \tilde{A}_{\alpha\beta}}{\partial \boldsymbol{Q}}&\Big(\mathbf{u}{,_1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \mathbf{u}{,_2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1)\Big) \\ &+ \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha{,}\beta}}{A}\Big((\boldsymbol{A}_3 \times \mathbf{u}{,_1}) \cdot \frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}} + (\mathbf{u}{,_1} \times \boldsymbol{A}_2) \cdot \frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}}\Big) \\ &+ \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha{,}\beta}}{A}\Big((\mathbf{u}{,_2} \times \boldsymbol{A}_3) \cdot \frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}} + (\boldsymbol{A}_1 \times \mathbf{u}{,_2}) \cdot \frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}}\Big), \end{aligned} \tag{112}$$

with:

$$\frac{\partial \tilde{A}_{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{\partial A^{-1}}{\partial \boldsymbol{Q}}\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha{,}\beta} + \frac{1}{A}\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}} \cdot \boldsymbol{A}_{\alpha{,}\beta} + \frac{1}{A}\boldsymbol{A}_3 \cdot \frac{\partial \boldsymbol{A}_{\alpha{,}\beta}}{\partial \boldsymbol{Q}}. \tag{113}$$

We already know how are expressed the derivatives of the covariant basis vectors $\partial \boldsymbol{A}_\alpha/\partial \boldsymbol{Q}$ [see Eq. (70)] and the derivatives of the Jacobian $\partial A/\partial \boldsymbol{Q}$ (see Eq. (105)). Nonetheless, there are some additional derivatives that are involved in the differentiation of the bending strains. Regarding the inverse of the Jacobian, the derivatives read as:

$$\frac{\partial A^{-1}}{\partial \boldsymbol{Q}} = -\frac{1}{A^2}\frac{\partial A}{\partial \boldsymbol{Q}}. \tag{114}$$

The derivatives of the director vector appears multiple times. After few developments, one should obtain the following formula:

$$\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}} = -\frac{1}{A}\boldsymbol{A}_3\frac{\partial A}{\partial \boldsymbol{Q}} + \frac{1}{A}\left(\frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}} \times \boldsymbol{A}_2 + \boldsymbol{A}_1 \times \frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}}\right). \tag{115}$$

Scalar products between these derivatives $\partial \boldsymbol{A}_3/\partial \boldsymbol{Q}$ and different vectors are involved. Let us give a general expression of this type of quantities:

$$\boldsymbol{v} \cdot \frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}_a} = -\frac{\boldsymbol{v} \cdot \boldsymbol{A}_3}{A}\frac{\partial A}{\partial \boldsymbol{Q}_a} \, - \, \frac{\boldsymbol{v} \times \boldsymbol{A}_2}{A}R_{a,1} \, - \, \frac{\boldsymbol{A}_1 \times \boldsymbol{v}}{A}R_{a,2}\,, \tag{116}$$

where $\boldsymbol{v}$ denotes any required vector. Lastly, let us give the following results:

$$\boldsymbol{v} \cdot \frac{\partial \boldsymbol{A}_{\alpha{,}\beta}}{\partial \boldsymbol{Q}_a} = R_{a,\alpha\beta}\,\boldsymbol{v}. \tag{117}$$

We now have all the ingredients in order to compute the derivatives of the bending strains *w.r.t.* the control points of the analysis model. It contains quite a lot of terms. Hence, it is worth spending some time to identify repetitive terms in order to make numerical savings in the implementation. For instance, there are multiple cross and dot products that seem

Table 1 Different types of gradients for the plate with a square hole as given in Fig. 2

Type	df_c/dx_1	df_c/dx_2	df_c/dx_3	df_c/dx_4	df_c/dx_5	df_c/dx_6
FD (1e-4)	-8.91674872	-2.90020134	-2.75637409	2.75637409	2.90020134	8.91674872
FD (1e-6)	-8.91674878	-2.90020142	-2.75637401	2.75637402	2.90020121	8.91674871
sAN (1e-6)	-8.91675602	-2.90020546	-2.75638323	2.75636483	2.90019719	8.91674136
sAN (1e-8)	-8.91675076	-2.90020354	-2.75637295	2.75636794	2.90019289	8.91674717
AN (-)	-8.91674869	-2.90020130	-2.75637405	2.75637405	2.90020130	8.91674869

The analysis model counts 8 × 8 quadratic elements. Note: values are magnified by 10^4

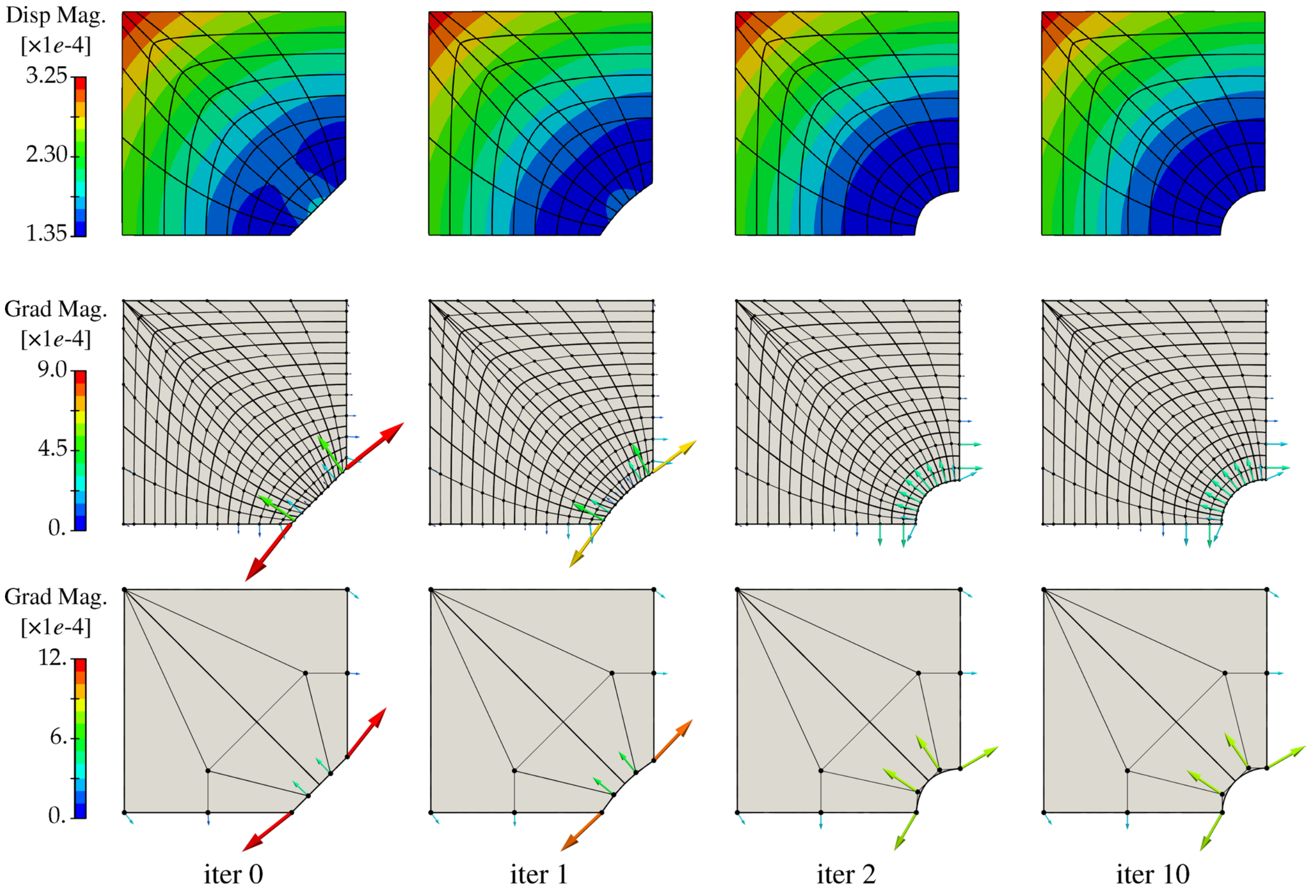

Fig. 4 History of structural analysis and sensitivity analysis during the shape optimization of the plate with a hole

better to compute once and for all at the beginning instead of computingthem for each control points $\boldsymbol{Q}_a$.

Finally, the derivatives of the bending moments are computed through the derivation of the constitutive Eq. (95):

$$\frac{\partial \boldsymbol{m}^{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{t^3}{12}\frac{\partial C_0^{\alpha\beta\gamma\delta}}{\partial \boldsymbol{Q}}\boldsymbol{\kappa}_{\gamma\delta} + \frac{t^3}{12}C_0^{\alpha\beta\gamma\delta}\frac{\partial \boldsymbol{\kappa}_{\gamma\delta}}{\partial \boldsymbol{Q}}. \tag{118}$$

At this point, we know how to compute all terms, i.e. the derivatives of the material tensor and the derivatives of the bending strains.

Implementation Regarding the implementation of the partial derivatives of the adjoint work *w.r.t.* the control point coordinates in case of the Kirchhoff–Love shell formulation, it is done similarly than for the standard solid element. Thus, we refer the interested reader to Algorithm 2 to get a global view of how it can be implemented.

5 Numerical Investigation

Now, we present the results obtained for the different examples already mentioned in Sect. 2.2. The goal is to collect a large range of benchmark results (spanning various structural analyzes and objective functions) which could be of interest during the development of new methods in the context of isogeometric shape optimization. The focus is on the sensitivities. For each example, we display the gradients and we give detailed values in tables. The gradients are represented by 3D fields of arrows. On the fine analysis models, we depicted the quantities denoted $\partial \mathrm{W}/\partial \boldsymbol{Q}$ throughout this document. On the design models, we plot the quantities $\partial \mathrm{W}/\partial \boldsymbol{P}$ which are obtained after the first propagation step. Let us mention that these quantities do not depend on the shape parametrization: even if one use different shape parametrizations than those of this work, one could rely on the presented results. For several examples, we give in tables detailed values of the full sensitivities (which depend on the shape parametrization). We verify the correctness of the presented analytical sensitivities in comparison with approximated ones. We also discuss the numerical efficiency of these sensitivities. Let us mention that we use, in this work, the SLSQP solver available in the NLopt library as the gradient-based algorithm to solve the optimization problems[44, 53].

5.1 Compliance as the Objective Function

Autoadjoint Problem The compliance seems to be the most common choice in structural optimization. By minimizing the compliance, the structure becomes stiffer in the sense that it deforms less. The compliance is a special case where the adjoint solution can directly be inferred from the state solution. In fact, the partial derivatives of the compliance (6) *w.r.t.* the design variables and the displacement DOF are respectively given by:

$$\frac{\partial f_c}{\partial x_i} = \frac{1}{2}\frac{\partial \boldsymbol{F}}{\partial x_i}\cdot \boldsymbol{u}, \tag{119}$$

$$\frac{\partial f_c}{\partial \boldsymbol{u}} = \frac{1}{2}\boldsymbol{F}. \tag{120}$$

Thus, the adjoint problem (15) reads, in the case of the compliance, as:

$$\mathbf{K}\boldsymbol{u}^* = \frac{1}{2}\boldsymbol{F}, \quad \Rightarrow \quad \boldsymbol{u}^* = \frac{1}{2}\boldsymbol{u}. \tag{121}$$

For the three examples tackled in this section, the load vector do not depend on the design variables. Thus, during the sensitivity analysis, one can omit the terms involving the differentiation of the load vector. If not, the partial derivatives of the compliance *w.r.t.* the design variables is computed firstly on the analysis model and then pull back the design variables level as explained in Sect. 3 [see more specifically Eq. (36)]. Here, we choose to omit them and to only consider the derivatives of the adjoint internal work during the sensitivity analysis.

Plate with a Hole The problem of the plate with a hole has been tackled multiple times in papers dealing with isogeometric shape optimization, see for instance Fußeder et al. [32], Hassani et al.[36], Qian[75] and Wall et al.[87]. The plate is subjected to a bi-axial loading. Initially, the shape of the hole is a square. It is known that the optimal shape for this problem consists in a circular hole, see Wall et al. [87]. The settings for this problem are given in Fig. 2. Due to symmetry, only one quarter of the plate is considered. Plane strain state is assumed. The optimization model is built using a single NURBS surface with 2 × 1 quadratic elements. The weights are set such that the circular hole can be exactly described as done in Qian[75]. We consider several refinement levels to define the analysis model.

More precisely, we firstly discretize the analysis model with 8 × 8 quadratic elements. Figure 4 shows several shape updates. For each shape update, we depict the solution of the structural analysis, and the gradients of the compliance at both analysis and design levels. The algorithm requires about 10 iterations to recover the optimal shape. In Table 1, we compare the analytical sensitivities (AN) with approximated sensitivities. The results are given for the initial geometry (i.e. the square hole). More specifically, we give in Table 1 the sensitivities obtained by global Finite Differences (FD) and the sensitivities obtained by semi-analytical approximation (sAN) as done, for example, in Kiendl et al. [50]. This sAN sensitivity consists in involving a finite difference scheme to approximate the derivatives of the element operators in Eq. (16). In Table 1, a forward scheme is used. We vary the perturbation step in order to highlight its influence on the sensitivities (see again Table 1). The first point that we want to underline is the correctness of the presented analytical sensitivities. In fact, we recover the FD results where there is likely no implementation errors due to simplicity. The differences arise only after a certain precision. These differences come from the approximation scheme. In fact, by varying the perturbation step, it is clear that only the first decimals of the approximated sensitivities are correct. This is especially true for the sAN sensitivities. The inaccuracy of the sAN scheme is well known and strategies for getting exact semi-analytical sensitivities have been proposed for standard FEM[7, 12, 31, 71, 85, 88]. Let us mention that one can use a central difference scheme in order to obtain a better accuracy of the approximated sensitivities. Eventually, one can observe that with the analytical sensitivities the following relations are obtained:

Table 2 Values of the gradients represented in Fig. 5. It highlights the influence of the refinement level of the analysis model on the sensitivity analysis. Note: values are magnified by 10^4

Refinement	df_c/dx_1	df_c/dx_2	df_c/dx_3	df_c/dx_4	df_c/dx_5	df_c/dx_6
$r = 1,\ p = 2$	− 7.646	− 3.916	− 3.600	3.600	3.916	7.646
$r = 2,\ p = 2$	− 8.917	− 2.900	− 2.756	2.756	2.900	8.917
$r = 3,\ p = 2$	− 9.671	− 2.236	− 2.170	2.170	2.236	9.671
$r = 1,\ p = 3$	− 8.641	− 3.134	− 2.966	2.966	3.134	8.641
$r = 2,\ p = 3$	− 9.483	− 2.406	− 2.318	2.318	2.406	9.483
$r = 3,\ p = 3$	− 9.995	− 1.941	− 1.909	1.909	1.941	9.995

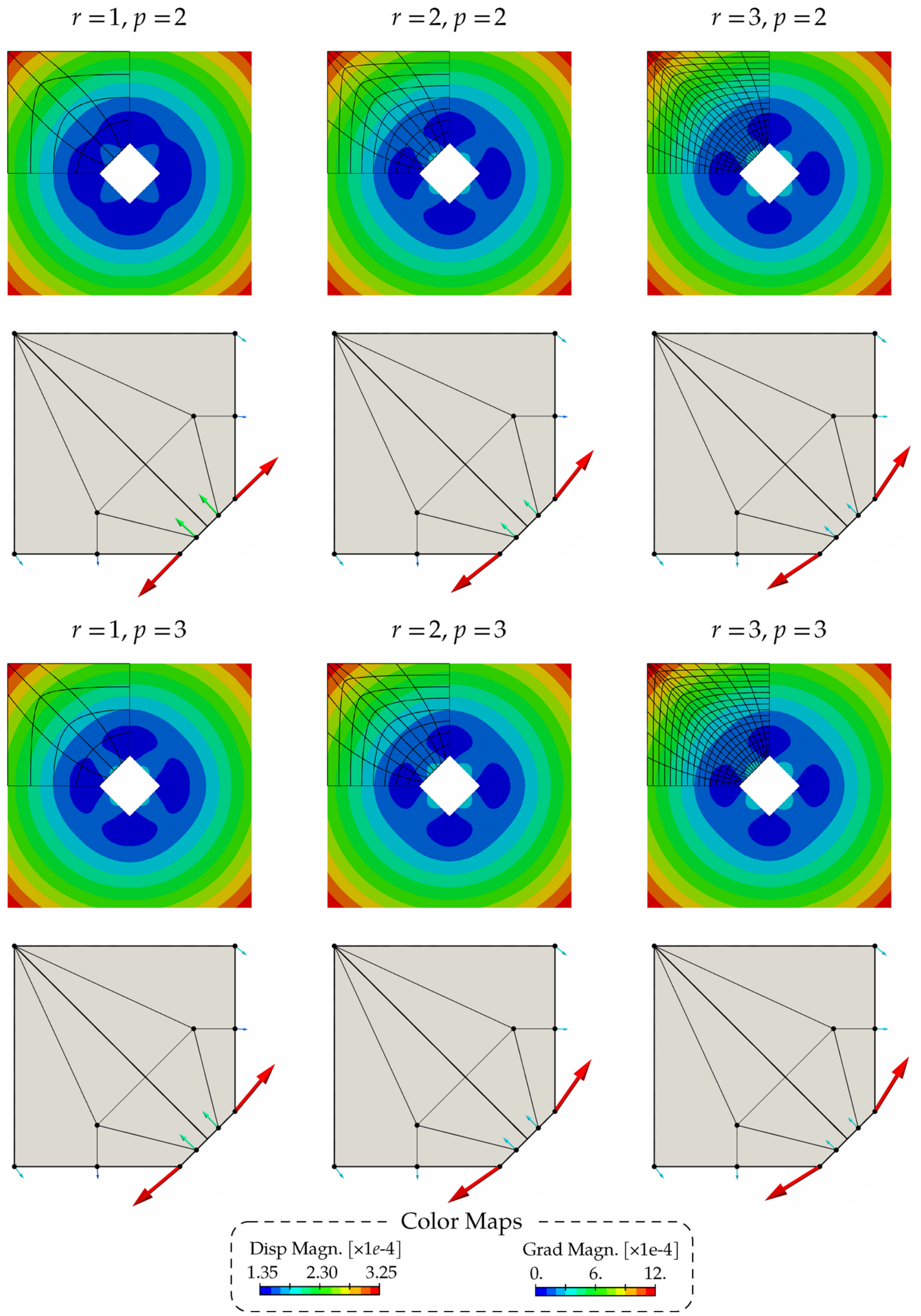

Fig. 5 Influence of the refinement level of the analysis model on the analytical sensitivity for the plate with a hole (r stands for the refinement level performed by knot insertion such that the analysis model has 2^r times more elements per direction than the design model)

Table 3 Influence on the sensitivities when every control point of the analysis model are used during the sensitivity analysis with the case where only the boundary control points are considered

Type	df_c/dx_1	df_c/dx_2	df_c/dx_3
All CPs (square)	− 8.917	− 2.900	− 2.756
Bound CPs (square)	− 9.278	− 3.027	− 2.578
All CPs (circular)	− 4.027	− 6.447	− 4.365
Bound CPs (circular)	− 4.054	− 6.498	− 4.396

Results are given for the plate with either a square hole and a circular hole. The analysis model counts 8 × 8 quadratic elements. Note: values are magnified by 10^4

$$\frac{df_c}{dx_1} = -\frac{df_c}{dx_6}, \qquad \frac{df_c}{dx_2} = -\frac{df_c}{dx_5}, \qquad \frac{df_c}{dx_3} = -\frac{df_c}{dx_4}.$$

It is clear that the exact sensitivity verifies these equations due the symmetry of the problem. The fact that we recover these equations demonstrates the good accuracy of the analytical sensitivity. This is not the case for the FD or sAN sensitivities (except FD with the perturbation step equal to 1e−4).

Regarding the accuracy of the analytical sensitivity, we study in Fig. 5 and Table 2 the influence of the refinement level of the analysis model. By looking at the presented results (Fig. 5 and Table 2), one can notice that the sensitivities are greatly affected by the refinement level. This is especially true for the initial configuration with the square hole due to the singularities at the corners of the hole. It means that the choice of the analysis model should not be made solely to ensure good results during the structural analysis but also to ensure good sensitivities.

Finally, it is very interesting to point out that only the control points associated to the domain boundary seem to give non-zero values in the gradients: see Fig. 4. In fact, all the interior control points lead to small values in $\partial \mathrm{W}/\partial \boldsymbol{Q}$ and $\partial \mathrm{W}/\partial \boldsymbol{P}$ in comparison with the boundary control points. The reason is that the interior control points do not modify the physical domain (from a continuum point of view). Thus, ideally speaking the influence of their positions on the internal and external works is null (as long as no geometrical singularities are introduced as overlaps etc.). Due to numerical errors in the analysis, this is not exactly observed. Indeed, it is known that mesh distortions impact the quality of the analysis in FEM-based simulations. However, one can consider during the sensitivity analysis to only compute the terms coming from the boundary control points and set to zero all the others associated to the interior control points. Instead of computing $\partial \mathrm{W}/\partial \boldsymbol{Q}$, one can compute the quantity $\partial \mathrm{W}/\partial \tilde{\boldsymbol{Q}}$ where the components are given by:

$$\frac{\partial \mathrm{W}}{\partial \tilde{\boldsymbol{Q}}_a} = \begin{cases} \dfrac{\partial \mathrm{W}}{\partial \boldsymbol{Q}_a} & \text{if } \boldsymbol{Q}_a \text{ moves the boundary,} \\ \mathbf{0} & \text{else.} \end{cases} \tag{122}$$

Making this choice enables to reduce the numerical cost of the sensitivity analysis. Instead of performing a loop on every element plus a loop on every active control point (as explained in Algorithm 2), we only need to integrate over the support of the basis functions associated to the boundary control points and to compute partial derivatives *w.r.t.* to these control points only. In practice, using $\partial \mathrm{W}/\partial \boldsymbol{Q}$ or $\partial \mathrm{W}/\partial \tilde{\boldsymbol{Q}}$ does not lead exactly to the same gradient, in general (again due to discretization errors). Computing all the terms (i.e. using $\partial \mathrm{W}/\partial \boldsymbol{Q}$) give the same gradient than the one obtained with Finite Differences as already shown in Table 1. When only the boundary control points are considered during the sensitivity analysis, the result is slightly different.

Table 3 highlights this issue: we consider the two calculation methods for the initial configuration (square hole) and the optimal configuration (circular hole). The difference is quite important, especially for the square hole due to the singularities in the solution which are badly captured with the chosen analysis model (8 × 8 quadratic elements). The discretization error is surely important here. With a finer analysis model, the influence of using either every control points or only the boundary control points, becomes much lower. With an analysis model with 64 × 64 cubic elements, the sensitivities for the plate with a circular hole are identical up to the sixth decimals. Thus, both approaches lead to the same optimal shape. Finally, this remark regarding the influence of interior control points on the sensitivity analysis is very related to the question of how to update the position of these control points during the optimization. Depending on the mesh density of the design model, the interior control points need to be moved in order to prevent the appearance of geometrical singularities due to element overlaps etc. This issue is beyond the scope of this article, and the interested reader is referred to[63, 77]. But let us mention that for each example presented in this work, we achieve to formulate shape parametrizations that automatically ensure a correct regularity of the geometries. Moreover, in what follows, we always consider every control points during the sensitivity analysis.

Square Shell Roof The optimization problem of the square shell roof is presented in Fig. 6. This example can be found in Hirschler et al.[38], and similar problems have been tackled by Bletzinger et al[11] and Kegl and Brank[47], for example. The roof is fixed at its corners and subjected to a vertical load (given by unit of area). The loading will not change with the shape update of the roof. The load vector can be built from the case of a square plate subjected to

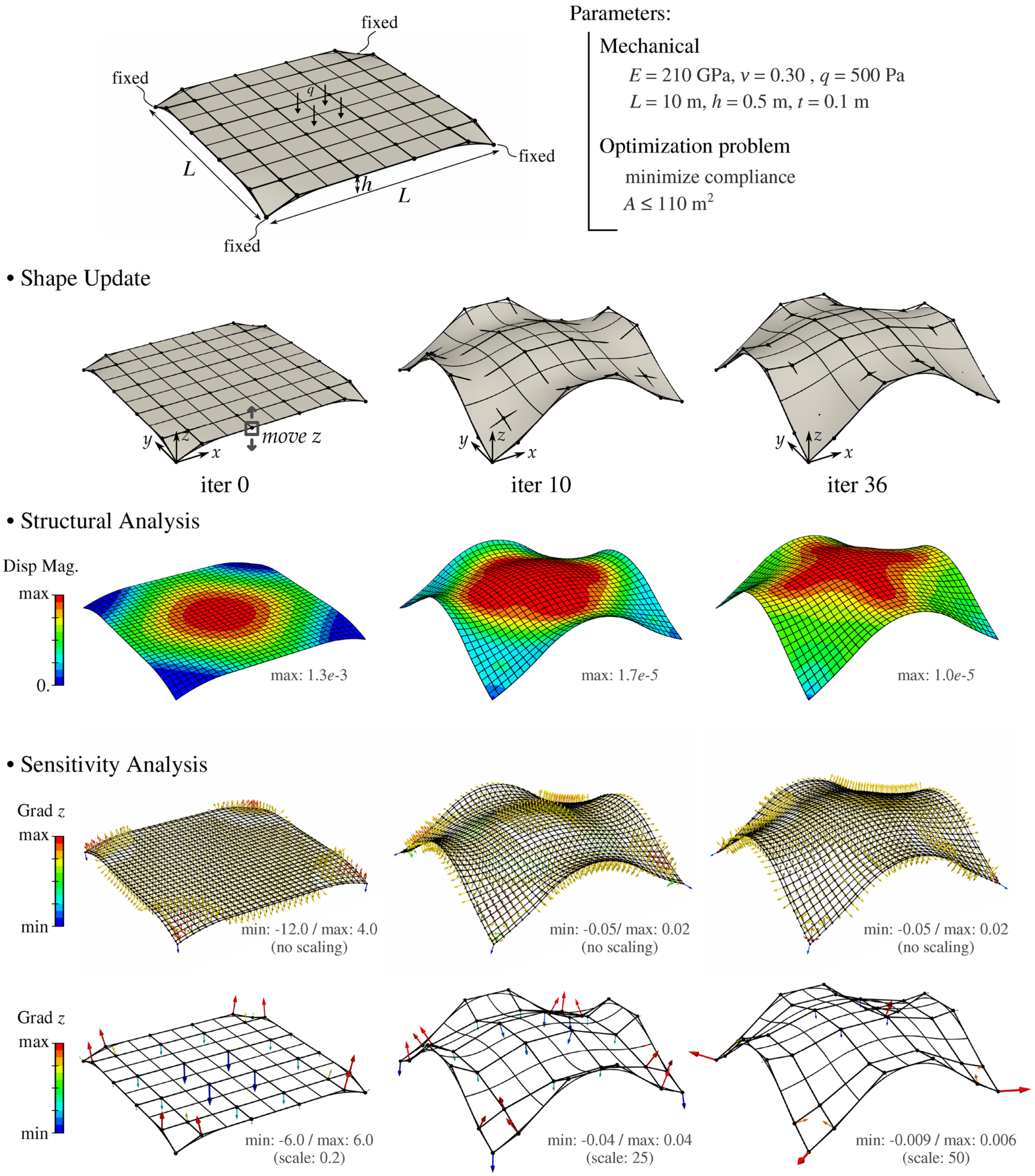

Fig. 6 Settings and optimization results for the square roof problem

uniform pressure. The same idea can be found for the optimal arch problem in Kiendl et al.[50].

The results presented in Fig. 6 and Table 4 are obtained with a design model with 4 × 4 quadratic B-Spline elements and an analysis model with 32 × 32 quadratic B-Spline elements. The Kirchhoff–Love shell formulation is used here to model the behavior of the roof. The shape parametrization consists in moving the control points of the design model in the z-direction (see Fig. 6). The control points located at the corners of the roof are left fixed. Thus, it leads to a total of

Table 4 Different types of gradients for the square roof as given in Fig. 6 (initial configuration)

Type (step)	df_c/dx_1	df_c/dx_2	df_c/dx_6	df_c/dx_7	df_c/dx_{13}	`Relative time`
FD (1e-5)	5.32678865	-1.25187196	2.77338596	-3.13803799	-5.83218544	70
central FD (1e-5)	5.32680063	-1.25171055	2.77340075	-3.13786987	-5.83145336	140
sAN (1e-5)	5.32295630	-1.25565732	2.76867265	-3.14312596	-5.83545230	30
central sAN (1e-5)	5.32680133	-1.25170945	2.77340078	-3.13787073	-5.83145388	60
AN (-)	5.32680133	-1.25170943	2.77340078	-3.13787073	-5.83145387	1

The 5 given values correspond to the unique values out of the 32 components of the gradients (Note: the design variables were numbered in a structured way starting with direction *x*)

32 design variables. Looking at the roof problem, one can notice that it has several symmetries. Consequently, the sensitivities have repeated terms; the influence of some design variables on the compliance is identical. For the initial geometry as given in Fig. 6, there are only 5 unique values in the sensitivity out of 32 components. These terms are given in Table 4. We compare the presented analytical sensitivity with several approximated sensitivities. More specifically, forward and central finite differential schemes are used to get either total FD sensitivities or sAN sensitivities. One can see that the forward scheme leads to quite significant differences with respect to the analytical sensitivity. Only the first two decimals are correct. With another perturbation step, the results can be somehow improved. However, this highlights one difficulty when employing approximated sensitivities: how to choose this perturbation step? Usually, a good point that limits its influence consists in scaling the problem such that the design variables vary between 0 and 1, for example. The objective functions and the constrains should also be scaled by using, for instance, their initial values. However, this does not guarantee that a given perturbation step will be suitable for every problem. It can be even trickier: a given perturbation can be suitable for the initial configuration but may lead to bad approximation of the sensitivities after some shape updates. More significantly, the adequate perturbation step (the one that leads to the lowest error) can be different for each design variable. But this is not identifiable in practice and usually one single perturbation step is chosen for each design variables and is kept the same during the whole optimization. It may be welcome to perform several sensitivity analyses for the initial configuration with different perturbation steps in order to select an appropriate one. For all the examples tackled in this work, this preliminary procedure was sufficient to limit the influence of the perturbation step: the optimization process always converged toward the same optimal shape and in a similar number of iterations when either approximated or analytical sensitivities were used. Finally, let us notice that even if the analytical sensitivity enables to get rid of the perturbation step, there is still the choice of the refinement level of the analysis model that needs to be done when setting up the optimization problem (see previous discussion for the plate with a hole). It can be interesting to adapt the refinement level during the optimization by using, for example, advanced tools as error estimators[2, 4, 14].

We also give in Table 4 the computational time for the different sensitivities. The computational times are scaled with the one of the analytical sensitivity. Unsurprisingly, the FD sensitivities takes the longest to compute because, for each design variable, system (3) needs to be built and solved. In case of the central FD scheme, this is even done twice per design variables. That is why the central FD takes twice the computational time of the forward FD (see again Table 4). Of course, this computational time can surely be reduced by saving redundant quantities and by using dedicated strategies as for example structural reanalysis[22, 52]. However, in this example, there are only 32 design variables. For more complex examples with thousands of design variables, FD sensitivities may be simply intractable[85]. Interestingly, the computational times of the sAN sensitivities are much higher than the AN sensitivity. Indeed, a factor roughly equal to the number of design variables (i.e. 32 here) is obtained when the forward finite difference scheme is used in the approximated part of the sAN sensitivity. Again, the computational time is doubled for the central sAN in comparison with the standard sAN (see again Table 4). The fact that the ratio between the computational time of the sAN and the AN sensitivities tends to the number of the design variables can be theoretically understood and could be inferred from Algorithm 2. The main calculation step of the AN sensitivity is the computation of the partial derivatives of the total adjoint work *w.r.t.* the control points coordinates $\partial \mathrm{W}/\partial \boldsymbol{Q}$ as previously explained. These terms take the form of an integral over the computational domain. In Algorithm 2, the steps that involve the most arithmetic operations are those from line 6 to line 10. The loop starting at line 12 consists essentially in adding terms to an array where every involved quantity is already computed. Thus, this loop goes fast. Interestingly, removing lines 12 to end of Algorithm 2 give the steps required to compute the total adjoint work (33). This means that the computation of the derivatives of the total adjoint work $\partial \mathrm{W}/\partial \boldsymbol{Q}$ takes a comparable amount of time than the computation of the total adjoint work in itself. Now, let us point out that during the sAN sensitivity analysis, one has to evaluate the total adjoint work as many times as design perturbations (i.e. as many times as the number of design variables with forward scheme, and twice as much with the central scheme). Because this step is the most time consuming step of the overall sensitivity analysis, it also means that the computation time of sAN is linearly proportional to the number of design variables. Let us summarize what has been observed:

$$\text{time(sAN)} \approx n_{\text{DV}} \times \text{time(W)}, \tag{123}$$

$$\text{time(AN)} \approx \text{time}(\partial \text{W}/\partial \boldsymbol{Q}) \approx \text{time(W)}. \tag{124}$$

This observation leads us to conclude that the time saving when using AN instead of sAN sensitivities is of the order of magnitude of the number of design variables:

$$\frac{\text{time(AN)}}{\text{time(sAN)}} \approx \frac{1}{n_{\text{DV}}}. \tag{125}$$

Also, the computational time of the AN sensitivity is (quasi) independent of the number of design variables. This is a great benefit of this method for the sensitivity analysis, especially when incorporating a large number of design variables.

Volumetric Beam The presented analytical sensitivity method can also be applied to 3D solid models. To highlight this point, we perform the shape optimization of a volumetric beam. The problem settings are described in Fig. 7. One end of the beam is fixed and an uniform pressure is applied over its top surface. The goal is to modify the cross section in order to maximize the stiffness of the structure. The final volume should not exceed the initial volume. Additional geometrical constrains are set in order to prevent undesirable shapes as described on Fig. 7. The design model is built using a B-Spline trivariate with 16 elements spread over the x-direction. Quadratic degree is taken in this particular direction whereas linear degree is taken in the two others. Thus, the design model counts $18 \times 2 \times 2$ control points. Only the 36 control points associated to the bottom surface (depicted in blue in Fig. 7) are movable. Two design variables are assigned to each of them. Thus, we end up with a total of 72 design variables. After refinement, the analysis model counts $32 \times 8 \times 8$ quadratic elements.

The results of the optimization are depicted in Fig. 7. The final geometry deforms much less which leads, in this case, to a significant reduction of the maximal Von-Mises stress (factor 25). We also give in Fig. 7 the sensitivities for several iterations of the resolution. Other isogeometric shape optimization of 3D examples can be found, for example, in

Blanchard et al.[6], Hassani et al.[36], Li and Qian[57], Lian et al.[59], and Wang et al.[91]. To the authors' knowledge, only Lian et al.[59] give quantitative results concerning the sensitivity analysis step. We hope that the presented example of the beam give additional useful results which could help to extend isogeometric shape optimization toward real-world applications.

5.2 Displacement as the Objective Function

In this section, we seek to use the displacement field as the objective function. For example, it can be interesting to adapt the shape of a structure such that it deforms in a prescribed manner. One application found in structural topology optimization using such objective functions is compliant mechanisms which are designed using such response functions[81, 98].

Adjoint Problem In the simple case described by Eq. (7), the partial derivatives of the objective function *w.r.t.* the design variables and the DOF read respectively as:

$$\frac{\partial f_{\text{u}}}{\partial x_i} = 0, \tag{126}$$

$$\frac{\partial f_{\text{u}}}{\partial \boldsymbol{u}} = \frac{1}{f_{\text{u}}} \frac{\partial \mathbf{u}^h}{\partial \boldsymbol{u}} \cdot \mathbf{u}^h. \tag{127}$$

Recalling the expression of the discrete displacement $\mathbf{u}^h$ [see again Eq. (7)], we get the expression of the missing derivatives involved in the right-hand side (127) of the adjoint problem:

$$\frac{\partial \mathbf{u}^h}{\partial \boldsymbol{u}_a} \cdot \mathbf{u}^h = R_a \mathbf{u}^h, \tag{128}$$

where R_a is the basis function associated to the displacement DOF of the control point numbered a. The basis functions are evaluated at the parameters ξ_M related to the physical point M where the displacement is computed. Here it is assumed that the point M is attached to the geometrical mapping. If not, the derivative (126) is not zero. Unlike the case of the compliance, taking f_{u} as the objective function requires the resolution of an additional system where the right-hand side is given by Eq. (127).

Planar Cantilever Beam The 2D cantilever beam problem depicted in Fig. 8 is derived from an already existing shape optimization problem that as been tackled within the IGA framework[67, 75, 87]. But instead of minimizing the compliance, we choose here to minimize the displacement at a specific position (see again Fig. 8). The initial geometry is a simple rectangle. The design region is the upper edge of the beam as described in Fig. 8. Here, the optimization model counts 4×1 elements. Degree 2 and 1 are taken in the directions x and y, respectively. It leads to 6 movable control points. We define 6 design variables that move the design control points in the y-direction. We set a lower bound to these design variables in order to prevent very thin sections.

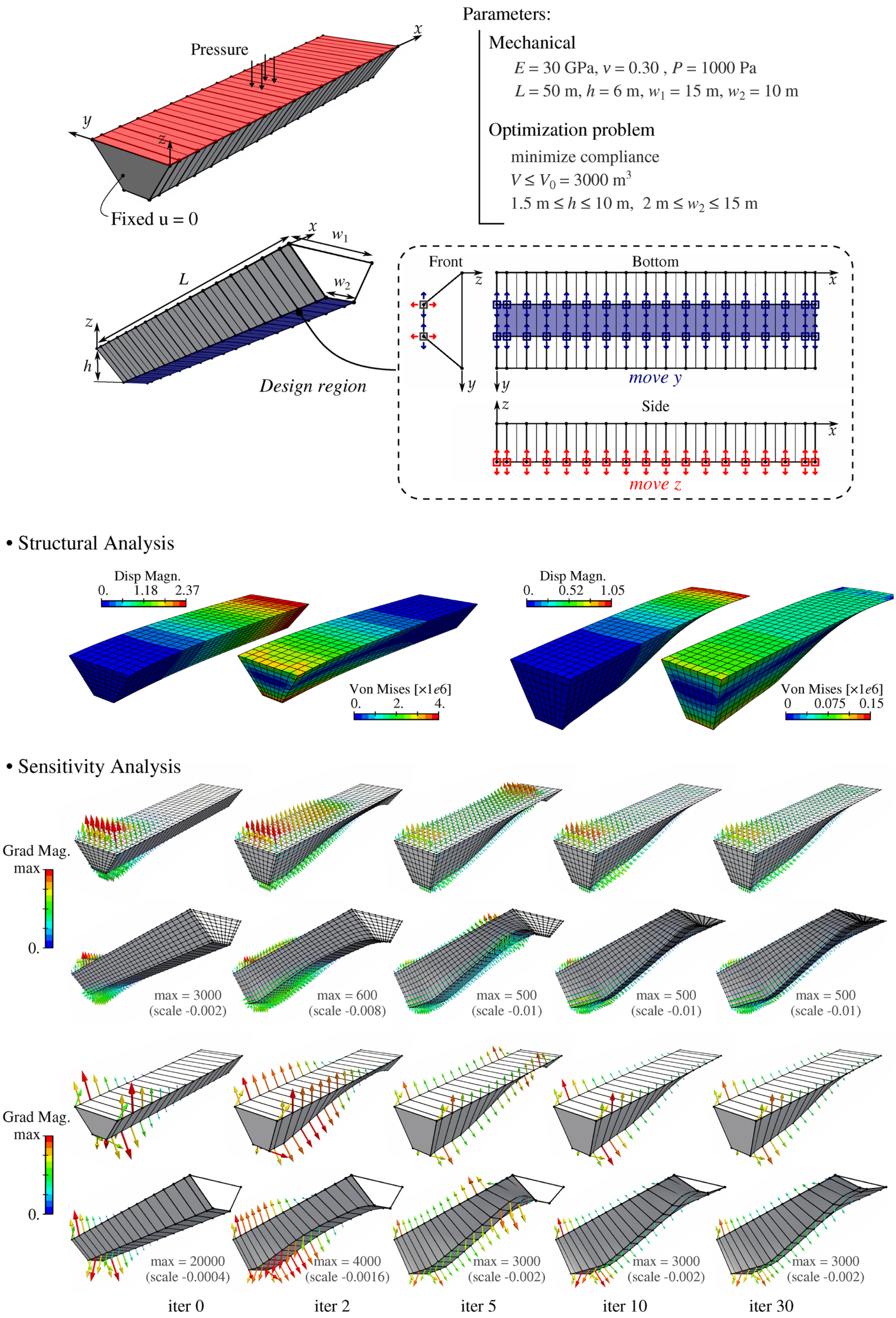

Fig. 7 Problem settings and results for the shape optimization of a 3D beam

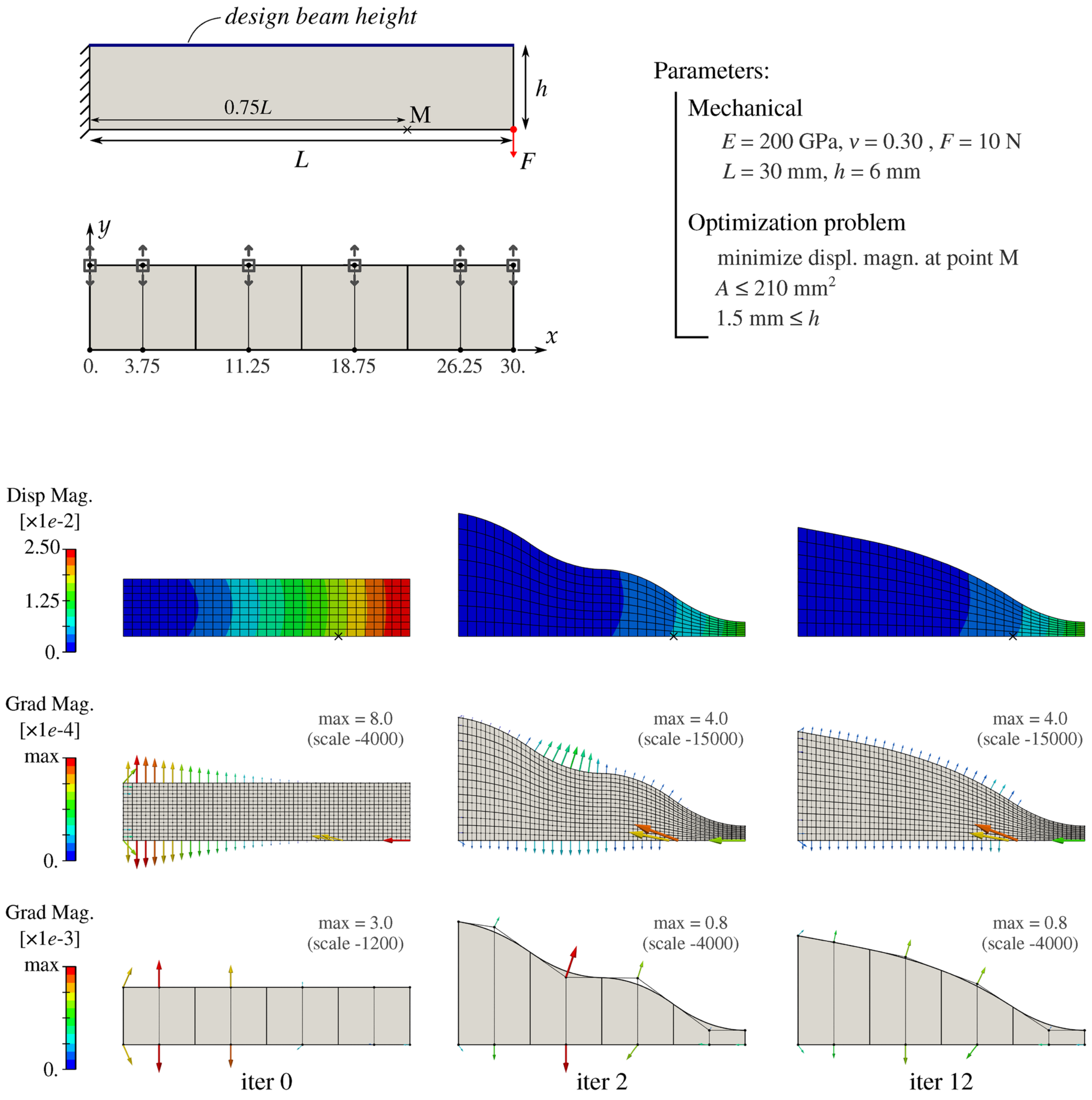

Fig. 8 Settings and optimization results for the cantilever beam problem

Moreover, the surface area should not exceed a given upper limit. All the settings are given in Fig. 8. The analysis model counts 32 × 8 quadratic elements. Additionally, plane strain state is assumed.

The optimization results for the planar cantilever beam problem are given in Fig. 8. In order to minimize the displacement at the specified location, the algorithm strengthens the portion of structure contained between the fixed side (where the Dirichlet boundary condition is applied) and the point M. On the contrary, the portion between the point M and the point of application of the load is weakened. It tends to limit the influence of the load: only a small part deforms a lot. It leads to an optimal shape where the upper edge is at first concave and then convex. As for all the examples investigated in this paper, we depict several results from the sensitivity analyses performed during the resolution. The gradients for the initial, an intermediary, and the final geometries are depicted in Fig. 8.

Roof with Minimal Deflection We perform a second example where the displacement field is directly used as the objective function. This example is identical to the square roof problem tackled in Sect. 5.1 and presented in Fig. 6. We make it new by only changing the objective function. Instead of minimizing the compliance, we undertake here to minimize the maximal deflection of the shell. To formulate such an objective, we call on a discrete P-norm (8). Differentiating the P-norm *w.r.t.* a quantity α reads as:

$$\frac{\partial \Phi}{\partial \alpha} = \left(\sum_{k=1}^{n_{pts}} \operatorname{sgn}(f_k) \frac{\partial f_k}{\partial \alpha} |f_k|^{P-1} \right) \left(\sum_{k=1}^{n_{pts}} |f_k|^P \right)^{1/P-1} . \tag{129}$$

By substituting Eqs. (126) and (127) into this last equation, one can get the expression of the analytical sensitivity of the P-norm approximation of the maximal displacement. The n_{pts} points where the displacement magnitude is evaluated in Eq. (129) are here the Greville points associated to the analysis model. Thus, there are as many evaluation points than control points of the analysis model; i.e. $n_{pts} = m_{cp}$.

The results for the roof problem with minimal deflection are given in Fig. 9. The results look very similar to the roof with minimal compliance but the final optimal shape slightly differs. This can be observed by comparing the final displacement fields in Figs. 6 and 9. Minimizing the P-norm (with $P = 20$ here) of the displacement magnitude leads to a maximal displacement equal to $||\mathbf{u}||_{\max} = 7.5 \times 10^{-6}$m. This is lower than the maximal displacement obtained when minimizing the compliance: it was equal to 10.0×10^{-6}m. On the contrary, the final shape has now a higher compliance in comparison with the results presented in Fig. 6. This highlight that the choice of the objective function (and the definition of the optimization problem in general) is crucial as it can lead to very unique designs. Finally, in Fig. 9 we display several results of the sensitivity analyses in order to give to the interested readers some quantitative data.

5.3 Stress Field as the Objective Function

Using the stress field to formulate objective functions and constraints is important to prevent the failure of the structure due to high local stresses.

Adjoint Problem As pointed out in Sect. 2.2, stress-based optimization is commonly done through the use of stress aggregation. In this work, we use the P-norm to track the maximal stress. As for the previous case involving the displacement as the objective function, using the stress field to express the objective function requires the resolution of an adjoint problem. The formulation of this adjoint problem for the objective function Φ_σ as given by Eq. (9) can be inferred by recalling the expressions of the discrete stresses (i.e. Eq. (57) for standard solid elements, and Eqs. (94) and (95) for Kirchhoff–Love shell elements) and the result from Eq. (129). The partial derivatives *w.r.t.* the design variables $\partial \Phi_\sigma / \partial x_i$ is performed firstly on the analysis model and then it is propagated to the design variable level as described by Eq. (36). One has to reuse the expression of the derivatives of the stress field *w.r.t.* the control point coordinates to get the full expression of the sensitivity (i.e. Eq. (75) for standard solid elements, and Eqs. (107) and (118) for Kirchhoff–Love shell elements). Finally, let us point out that the components of the stress field expressed in the covariant basis [as given in Eq. (48)] has no physical meaning. A change of basis into a (local) Cartesian basis is further required[48]:

$$\hat{\boldsymbol{\sigma}}^{ij} = \boldsymbol{\sigma}^{kl} \left(\boldsymbol{E}_i \cdot \boldsymbol{G}_k \right) \left(\boldsymbol{G}_l \cdot \boldsymbol{E}_j \right). \tag{130}$$

This change of basis needs to be taken into account during the sensitivity analysis:

$$\begin{aligned} \frac{\partial \hat{\boldsymbol{\sigma}}^{ij}}{\partial \boldsymbol{Q}} &= \frac{\partial \boldsymbol{\sigma}^{kl}}{\partial \boldsymbol{Q}} \left(\boldsymbol{E}_i \cdot \boldsymbol{G}_k \right) \left(\boldsymbol{G}_l \cdot \boldsymbol{E}_j \right) \\ &\quad + \boldsymbol{\sigma}^{kl} \left(\boldsymbol{E}_i \cdot \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{Q}} \left(\boldsymbol{G}_l \cdot \boldsymbol{E}_j \right) + \left(\boldsymbol{E}_i \cdot \boldsymbol{G}_k \right) \frac{\partial \boldsymbol{G}_l}{\partial \boldsymbol{Q}} \cdot \boldsymbol{E}_j \right). \end{aligned} \tag{131}$$

If the Cartesian basis is local, then the basis vectors $\boldsymbol{E}_i$ may also depend on the control point coordinates. Thus, their derivatives should be also taken into account in equation (131). The derivatives of the covariant basis vectors *w.r.t.* the control point coordinates have already been given, see Eq. (70).

Tensile Specimen Using the stress field in the objective function enables to reduce stress concentration. In order to highlight this point, we deal with the design of a tensile specimen as described in Fig. 10. It is a planar problem where plane stress state is assumed. The specimen is subjected to traction. The goal is to design the transition between the wide ends of the specimen and its thin central part. The initial design connects these sections with straight lines. It creates stress concentration due to the sharp transition zones in the design as shown in Fig. 10. Thus, to reduce these stress concentration zones, we parametrize the shape of the specimen (see again Fig. 10). Due to the symmetry of the problem, only one quarter of the structure is considered. More precisely, the optimization model contains three patches with C^0-continuity junctions. Only the subdomain denoted $\Omega^{(2)}$ in Fig. 10 is subjected to shape changes. This patch has degree 3 in the x-direction and degree 1 in the y-direction. It counts 4×1 elements. The shape parametrization consists in moving 5 of its control points in the y-direction as depicted in Fig. 10. The two other patches $\Omega^{(1)}$ and $\Omega^{(3)}$ are discretized using one linear single element each so that the patch interfaces are matching. The analysis model is built through k-refinement such that each

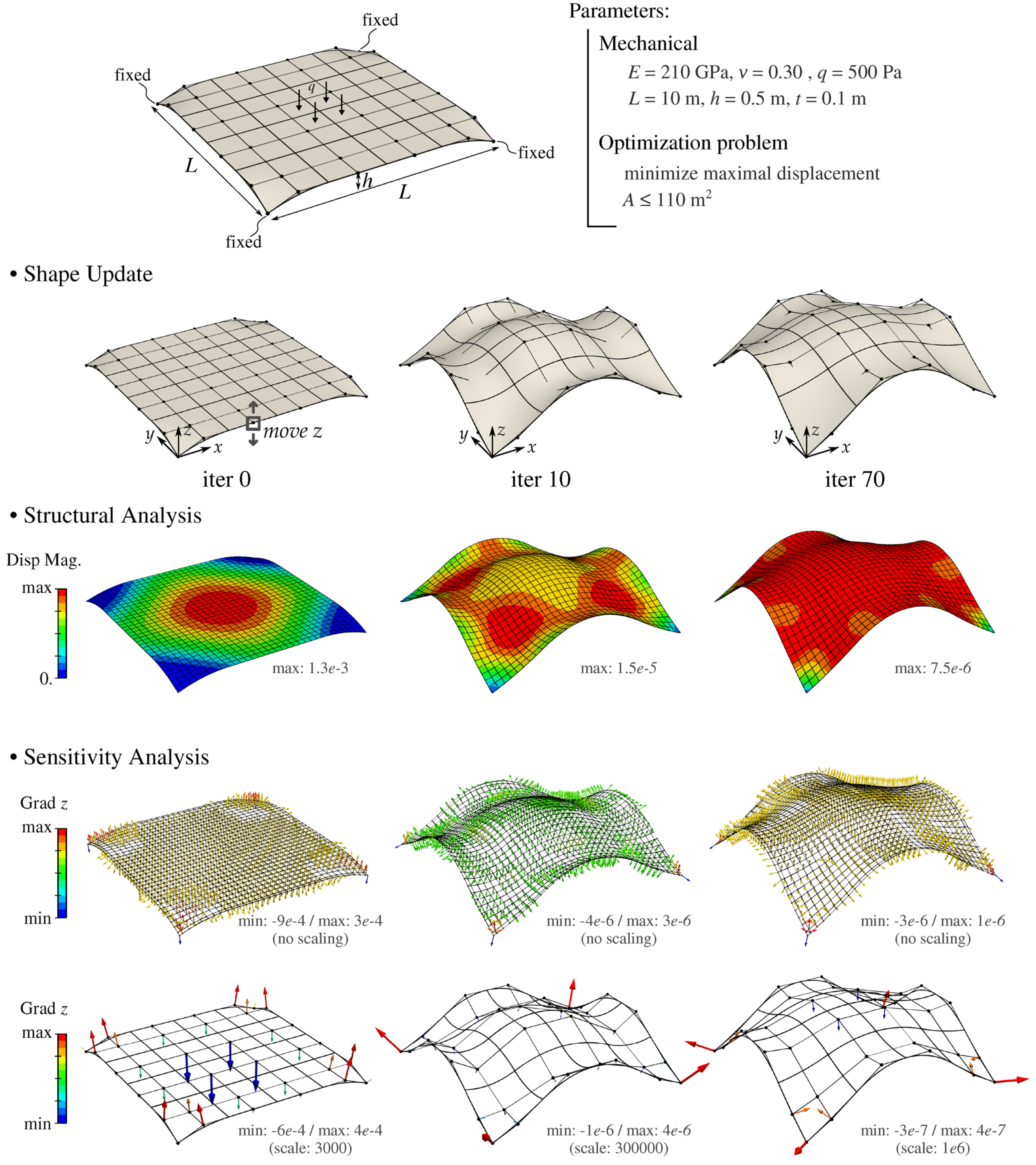

Fig. 9 Settings and optimization results for the square roof with minimal deflection problem

patch is discretized with 16 × 16 cubic elements. The goal of this optimization problem consists in minimizing the maximal von Mises stress occurring in the overall specimen. For plane stress state, the von Mises stress reads as:

$$\sigma_{\mathrm{VM}} = \sqrt{\hat{\sigma}^{11}\hat{\sigma}^{11} - \hat{\sigma}^{11}\hat{\sigma}^{22} + \hat{\sigma}^{22}\hat{\sigma}^{22} + 3\hat{\sigma}^{12}\hat{\sigma}^{12}}. \qquad (132)$$

The objective function is taken here as the *P*-norm of the von Mises stress σ_{VM} computed at the Greville abscissas of each B-Spline bivariates. Thus, the von Mises stress is

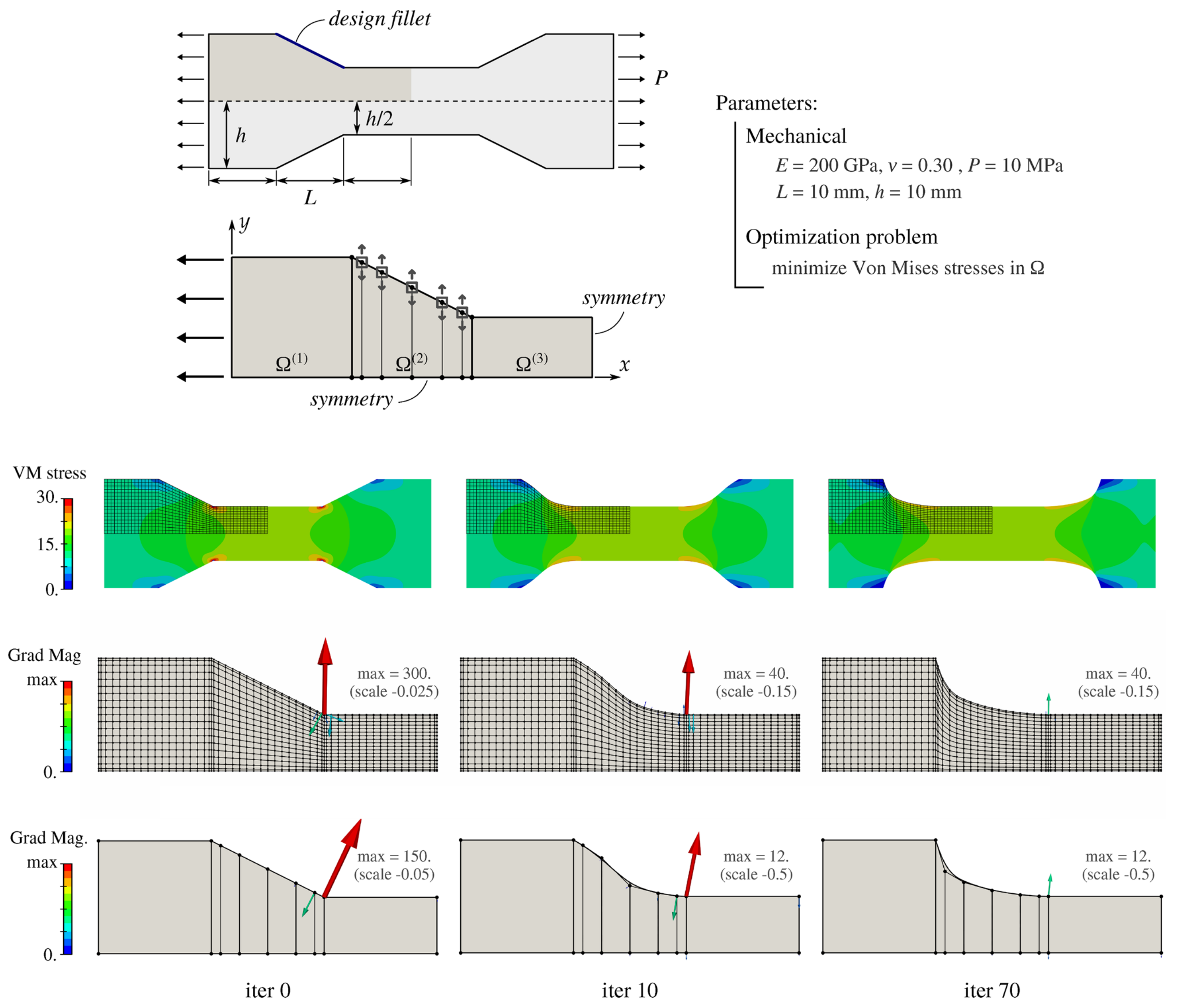

Fig. 10 Settings and optimization results for the tensile specimen problem

evaluated at $3 \times 19 \times 19 = 1083$ points. We take $P = 40$ for this example.

The results of the optimization of the tensile specimen are presented in Fig. 10. The final shape consists in a smooth fillet that leads to reduce the maximal von Mises stress from 51.1 to 22.7 MPa. We also give several sensitivity analyses in Fig. 10. Table 5 gives the values of the derivatives for the initial configuration. We compare the AN calculation with the approximated sensitivities (FD and sAN). As for the previous studies of the plate with a hole and the square shell roof, the AN sensitivity is close to the approximated sensitivities. It enables to validate its formulation and its correct implementation. Even if the objective function seems to be quite complex, full analytical sensitivity analysis is

Table 5 Different types of gradients for the tensile specimen as given in Fig. 10 (initial configuration)

Type (step)	df_c/dx_1	df_c/dx_2	df_c/dx_3	df_c/dx_4	df_c/dx_5
FD (1e-4)	0.04323446	0.22659049	1.45409336	6.65664040	44.13258751
central FD (1e-4)	0.04323668	0.22659784	1.45412855	6.65676127	44.13280705
sAN (1e-4)	0.04323577	0.22659459	1.45410643	6.65661209	44.13040686
central sAN (1e-4)	0.04323670	0.22659783	1.45412854	6.65676128	44.13280707
AN (-)	0.04323670	0.22659783	1.45412854	6.65676128	44.13280746

achievable with the present framework. Thus, our approach is versatile and enables to tackle a large variety of response functions. Finally, let us point out that similar optimization problems have been presented in the context of IGA, see for example Li and Qian[57], Wang and Turteltaub[90] and Wang et al.[91]. Especially, Wang and Turteltaub[90] detail how continuous adjoint sensitivity analysis can be performed in the context of isogeometric shape optimization.

Catenary Stress-based structural design can also be useful to design slender structures as shells. In this context, we are generally interested in limiting the bending deformations of these structures in order to guarantee their durability[8]. A shell is much stiffer in the in-plane directions than in the out-of-plane direction. Thus, an appropriate design objective for shells consists in minimizing the bending moments along the structure[11]. Again, we consider here to use the discrete P-norm (8) of the bending moments computed at the Greville abscissas (with $P = 40$). More specifically, we seek to solve the catenary problem as described in Fig. 11. The arch is fixed at its ends (no displacement) and is subjected to a distributed loading which simulates the effect of gravity. The objective is to minimize the bending moment along the y-axis. We also impose a geometric constraint: the arch length l should be equal to 15 m. Interestingly, this problem is related to well-known results of hanging chain or cable. When supported only at its ends and under its own weight, a chain assumes a specific shape known as a catenary curve[84, 94]. Here the problem is very similar and, finally, the optimal arch that minimizes the maximum bending moment would be described by a catenary (this is even more true when the thickness of the shell tends towards zero). Bandara and Cirak[4] performs a similar study but they use the compliance to formulate the optimization problem. There exists an analytical solution of the optimization problem described in Fig. 11. The height of the catenary arch is:

$$h_{\mathrm{AN}} = a \cosh\left(L/a\right) - a, \tag{133}$$

where a is the solution of:

$$2a \sinh\left(L/a\right) - l = 0. \tag{134}$$

The distance L is defined in Fig. 11. The catenary is then defined by the following equation:

$$y = -a \cosh\left(x/a - L/2/a\right) + a + h_{\mathrm{AN}}, \quad 0 \leq x \leq L. \tag{135}$$

The exact analytical solution cannot be completely recovered with a B-Spline design model since B-Splines cannot represent exactly an hyperbola. However, we should get an optimal shape that approximates the catenary. We use here a design model with quadratic degree in the principal arch direction and linear degree along the width. The design model counts 4 × 1 elements. We define 4 design variables that move couples of control points in the z-direction as depicted in Fig. 11. The analysis model is defined through k-refinement such that the discretization counts 32 × 2 cubic elements.

The optimization results are given in Fig. 11. The final shape obtained with the chosen design and analysis models is described. Even with the proposed coarse design model, the optimal shape looks like a catenary. More specifically, the optimal arch meets the analytical height h_{AN}. Also, the shape of the optimal arch is visually very close to a catenary. The difference is quasi indistinguishable. The bending moment is nearly zero for the final arch. The structure is subjected to membrane forces only. This was the goal of the optimization. We also give in Fig. 11 several sensitivity analyses. The presented example can be used by the interested reader to validate the implementation of the analytical sensitivity. Due to its simplicity, this example enables to test new methods. A similar example of an arch optimization where an analytical solution exists can be found in Kiendl et al.[50].

5.4 Objective Functions Involving Natural Frequencies

The last example, described in Fig. 12, further highlights the versatility of the present framework. Instead of solving a linear elasticity problem, we perform here a natural frequencies analysis. We thus consider objective functions of the form:

$$f \; := f(x, \Lambda(x)), \tag{136}$$

where the vector Λ collects the n_λ eigenvalues associated to the problem (4). Vector x collects the design variables as in Eq. (2).

Sensitivity Analysis for Natural Frequencies The sensitivity analysis of response functions involving natural frequencies is done in a very similar manner than in case of standard sensitivity analysis (i.e. involving the displacement). It is done accordingly to the process described in Fig. 1. Derivatives are firstly computed on the analysis model and then propagated back to the design variables level using the refinement operators and the shape parametrization.

We already have almost all the ingredients to perform the sensitivity analysis of response functions of the form (136). The total derivatives of a response function of the form (136) *w.r.t.* a design variable is given by:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \sum_{k=1}^{n_\lambda} \frac{\partial f}{\partial \lambda_k} \frac{d\lambda_k}{dx_i}. \tag{137}$$

By differentiating the governing equations (4), one get the derivatives of an eigenvalue λ *w.r.t.* a design variable:

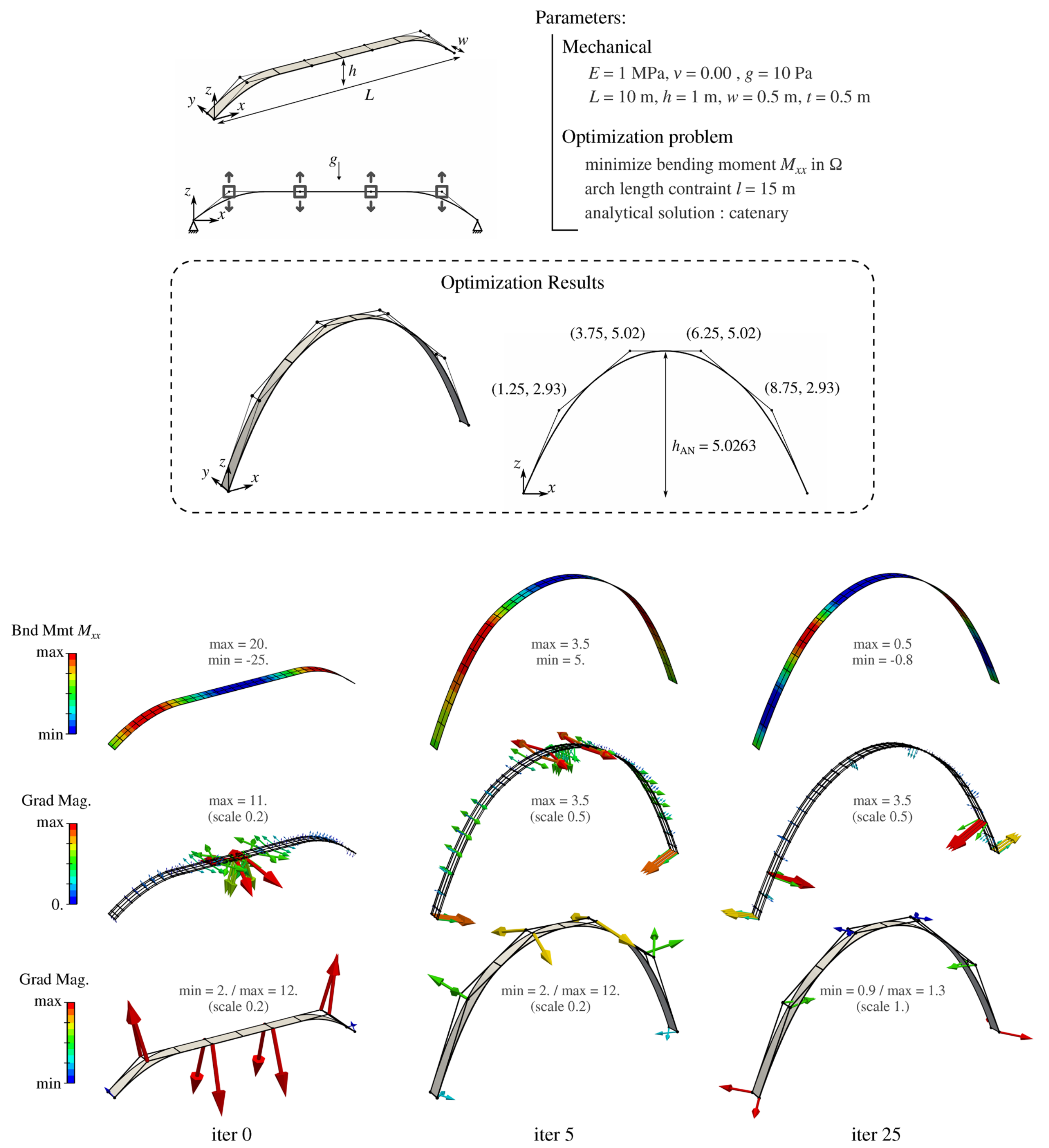

Fig. 11 Settings and optimization results for the catenary problem

$$\frac{d\lambda}{dx_i} = \frac{1}{\boldsymbol{v}^T \mathbf{M} \boldsymbol{v}} \left[\boldsymbol{v}^T \left(\frac{\partial \mathbf{K}}{\partial x_i} - \lambda \frac{\partial \mathbf{M}}{\partial x_i} \right) \boldsymbol{v} \right]. \tag{138}$$

The eigenvector $\boldsymbol{v}$ corresponding to eigenvalue λ is often normalized *w.r.t.* the mass matrix which gives $\boldsymbol{v}^T \mathbf{M} \boldsymbol{v} = 1$. We make this consideration in the rest of the document.

Then, we follow the same strategy than the one presented in Sects. 3 and 4. One would get the following expression:

$$\frac{d\lambda}{dx_i} = \frac{\partial \boldsymbol{P}}{\partial x_i} : \mathbf{R}^T \left(\frac{\partial \mathrm{W}_{\mathrm{dyn}}}{\partial \boldsymbol{Q}} - \frac{\partial \mathrm{W}_{\mathrm{int}}}{\partial \boldsymbol{Q}} \right), \tag{139}$$

where the dynamic work $\mathrm{W}_{\mathrm{dyn}}$ reads as:

$$\mathrm{W}_{\mathrm{dyn}}(\lambda, \mathbf{v}^h) = -\lambda \int_{\bar{\Omega}} \rho\left(\mathbf{v}^h \cdot \mathbf{v}^h\right) |J| \mathrm{d}\bar{\Omega}. \tag{140}$$

Under the consideration of constant density ρ, the derivatives $\partial \mathrm{W}_{\mathrm{dyn}} / \partial \boldsymbol{Q}$ are simply given by:

$$\frac{\partial \mathrm{W}_{\mathrm{dyn}}}{\partial \boldsymbol{Q}} = -\lambda \int_{\bar{\Omega}} \rho\left(\mathbf{v}^h \cdot \mathbf{v}^h\right) \frac{\partial |J|}{\partial \boldsymbol{Q}} \mathrm{d}\bar{\Omega}. \tag{141}$$

The discretized solution $\mathbf{v}^h$ is expressed as in Eq. (54) where the DOF are taken as the eigenvector. We already know how to get the derivatives of the internal work *w.r.t.* the control point coordinates [see Eq. (83)]. Regarding the implementation, the derivatives of the dynamic work (141) is performed as in Algorithm 2. Also, if multiple eigenvalues are involved in the response function then one should compute the derivatives $\partial \mathrm{W}_{\mathrm{dyn}} / \partial \boldsymbol{Q}$ and $\partial \mathrm{W}_{\mathrm{int}} / \partial \boldsymbol{Q}$ for each eigenpair simultaneously and not one after the other.

Elephant Trunk We named the novel example proposed in Fig. 12 as the elephant trunk problem due to the obtained optimal shape. The problem is as follows. The initial geometry is a plain cylinder with an ellipse-like cross section. The bottom face of the structure is fixed while the rest is free of motion. The first two vibration modes are shown in Fig. 12. We parametrize the shape of the structure by layers of control points in the z-direction. For each layer, we assign two design variables which modify the width of the cross section in the x- and y-direction, respectively. Again, the reader is referred to Fig. 12. The design model is built using a NURBS trivariate with 4 quadratic elements. It leads to 12 design variables. The refinement level for the analysis model is chosen such that it counts $4 \times 4 \times 16$ quadratic elements.

The goal of this optimization problem is to maximize the lowest natural frequency of the structure. Optimization problems involving an eigenvalue problem are known to be challenging due to mode switching and multiple eigenvalues. Without any particular treatment, the optimization algorithm may encounter difficulties in converging due to none differentiability of the eigenvalues[34, 62, 76]. This difficulty is encountered with the present example. In a naive way, one can try to run the optimization without any treatment: i.e. by using the objective function f_{λ_1} as defined in Eq. (11) and by taking λ_1 as the lowest eigenvalue at every iteration (no mode tracking). The convergence history is given in Fig. 13. The stopping criterion is defined as the relative difference in the objective function between two successive iterations. The algorithm takes almost 100 iterations to converge. Figure 13b provides a closer look to what happens. Several mode switches occur during the optimization. The reason is the following. Let us imagine that at iteration k, the eigenvalue λ_x^k associated to the bending mode in the x-direction is the lowest. The result from the sensitivity analysis will then lead us to increase the width of the cross section in this particular x-direction. But due to the volume constraint, the algorithm reduces the width of the cross section in the other direction y. What happens is that, at the next iteration, the eigenvalue λ_y^{k+1} may become lower than the new eigenvalue λ_x^{k+1}. Even more, it is possible that the eigenvalue λ_y^{k+1} is lower than the initial eigenvalue λ_x^k. The algorithm is then confused; instead of increasing the lowest eigenvalue (which was λ_x^k), it has decreased it (the new λ_y^{k+1}). This is what happens at iteration 19 and causes the undesirable pic (see again Fig. 13b). When the optimal shape has multiple eigenfrequencies, the algorithm faces huge difficulties to converge if no particular treatment is handled. Regarding the vibration of structures, one would face these situations when the final optimal shape has some symmetries. Indeed, having geometrical symmetries yields the presence of multiple eigenvalues, which are known to initiate non-differentiability of the objective function as already pointed out.

Several methods can be employed to tackle this issue. As already mentioned, dedicated sensitivity analysis can be performed in the case of multiple eigenvalues[34, 62, 76, 95]. One can also reformulate the optimization problem by using the so-called bound formulation[62, 66, 70, 82]. Another possibility is to impose constraints to enforce mode separation[79, 80, 82]. Stanford et al.[82] compare both methods for the topology optimization of aerostructures. Here, we adopt a strategy similar to Manh et al.[63]. We reformulate the objective function as presented in Eq. (11). For this example, we only consider to first two modes in the definition of function $\boldsymbol{\Phi}_\lambda$ since the third mode is sufficiently higher so that it does not interact with them. When the coefficient P becomes high, the function $\boldsymbol{\Phi}_\lambda$ is a close upper approximation of f_{λ_1}. Here we choose $P = 40$. This function $\boldsymbol{\Phi}_\lambda$ has the advantage of being differentiable even if the first two modes are repeated modes. It leads to smooth convergence of the algorithm as shown in Fig. 13c. We end up with a geometry that has exact axial symmetric. The first two eigenvalues λ_x and λ_y are equal. Additionally, Fig. 12 presents the results from several vibration analyses and sensitivity analyses for this elephant trunk problems. We believe that this example, despite its apparent simplicity, enables to study the robustness of new approaches for the eigenfrequency optimization of structures. Other examples regarding the isogeometric shape optimization for natural frequency can be found, for example, in Lei et al.[56], Manh et al.[63] and Taheri and Hassani[83].

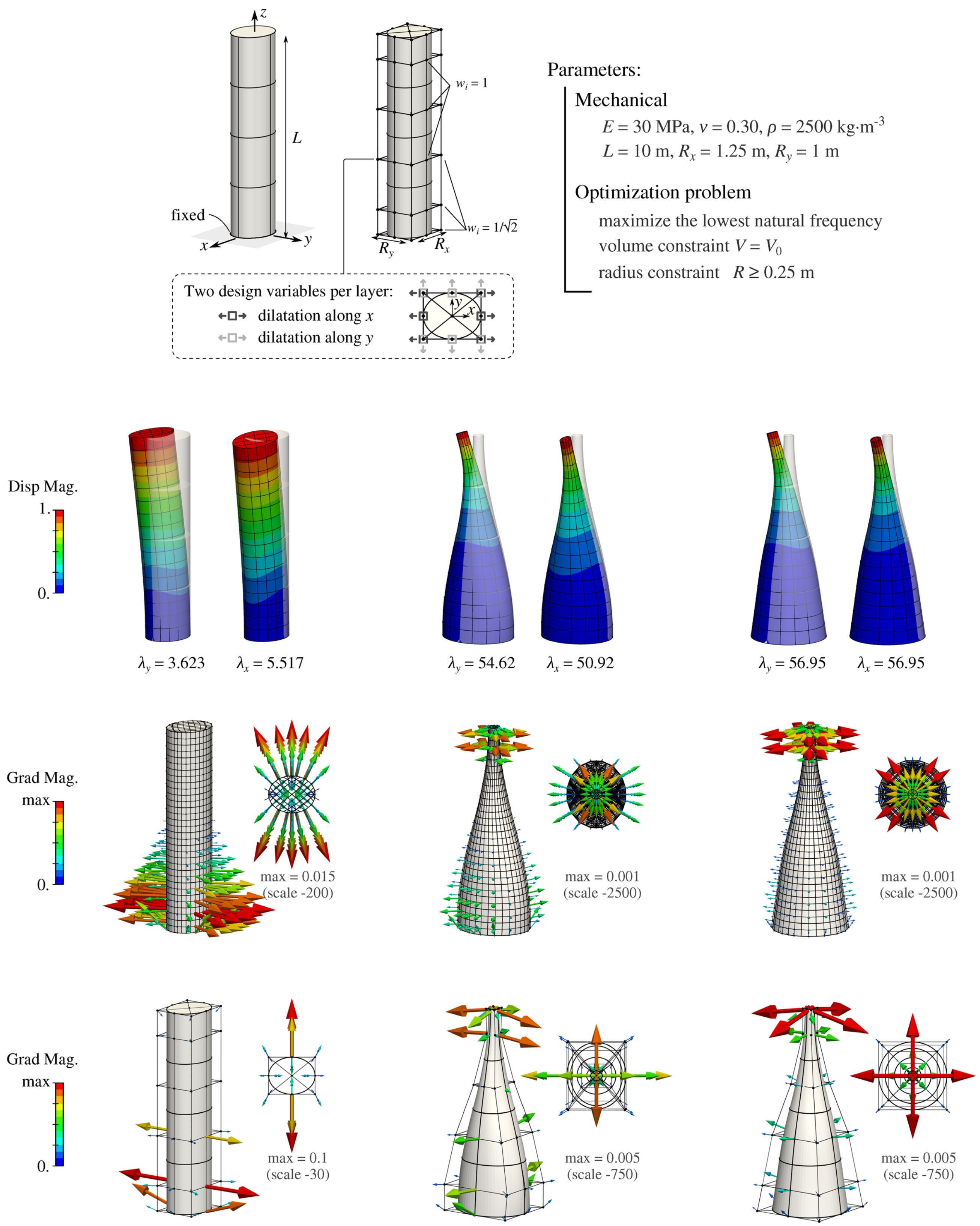

Fig. 12 Settings and optimization results for the elephant trunk problem

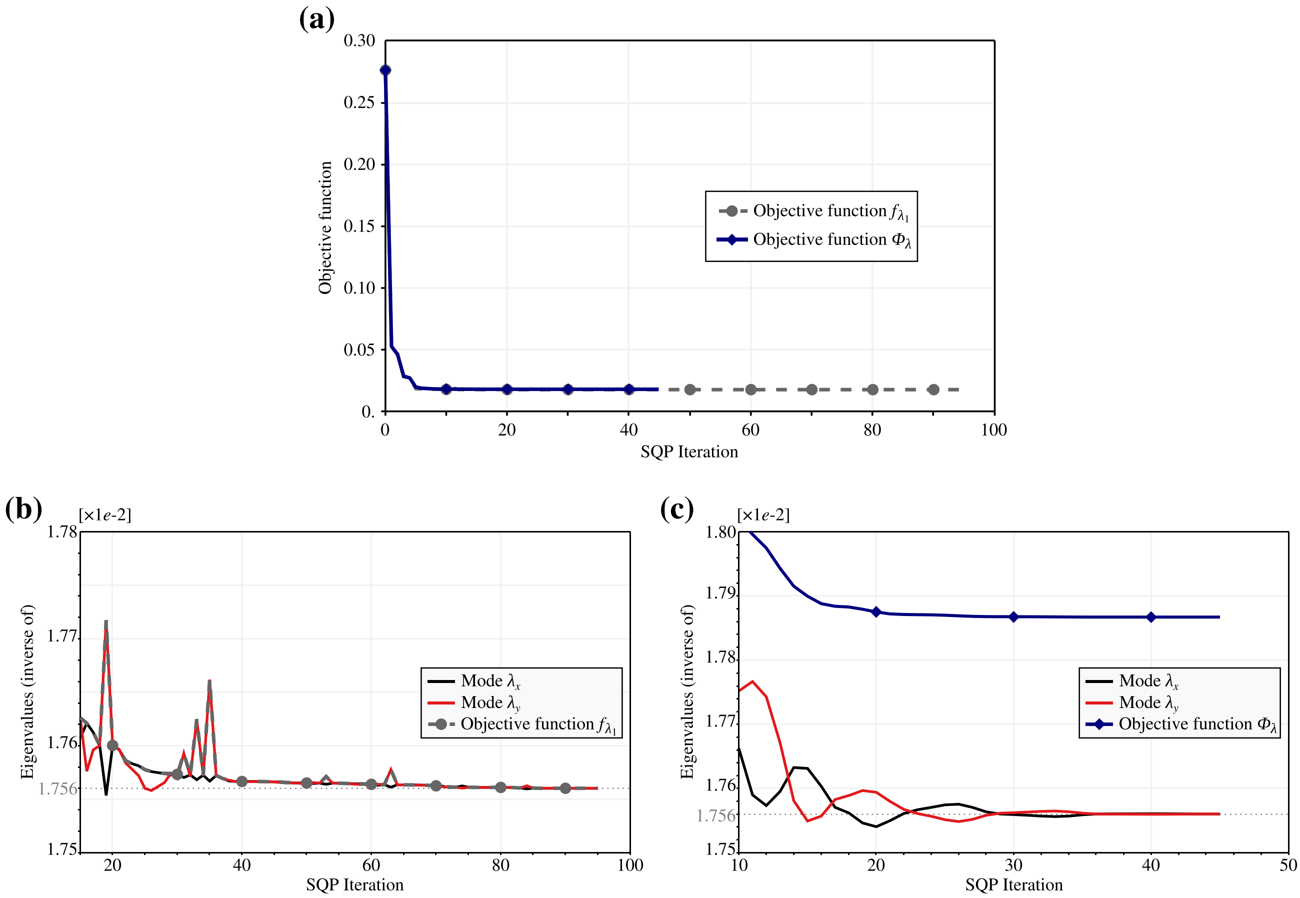

Fig. 13 Convergence histories for the elephant trunk problem with the two investigated objective functions: **a** global view, **b** closer look at the bad convergence when no treatment of the mode switches is done, and **c** better convergence is obtained with the aggregation by a *P*-norm

6 Conclusion

For each of the tackled examples, we observed a great benefit of employing analytical sensitivities especially in terms of computational time and accuracy. In fact, the presented approach was found to be (quasi) independent of the number of design variables. The main calculation step occurs on the analysis model where the derivatives of some scalar functions *w.r.t.* the control points are computed. Indeed, we found that some commuting steps enable to drastically reduce the size of the operators to be differentiated. Here, we did not derive one-by-one every components of the finite element matrices *w.r.t.* the design variables as it is often presented in the literature for discrete sensitivities. Instead, we viewed the required derivatives as the partial differentiation of the (discretized) weak formulation of the state equation. For instance, in the case of static analysis, we partially derived the internal and external works *w.r.t.* the control points of the analysis model. We observed that this step is eased by formulating the element using a curvilinear formalism (as commonly done with shells). Then, these gradients were mapped back to the coarser design model on which the shape parametrization was set. Finally, these gradients are brought at the level of the design variables using the definition of this shape parametrization. The propagation of the sensitivity from the analysis model to the design variables is achievable within the isogeometric framework because of the inherent tight link between the design model and the analysis model. The presented analytical sensitivity analysis works even for complex non-linear shape parametrizations.

More generally, this work undertook to formalize IGA-based shape optimization based on all the published works related with this issue. We highlighted the potential of this approach for the optimal design of structures. Surely, an efficient sensitivity analysis is of particular importance to build a robust framework which would be applicable for industrial applications. We believe that the new presented sensitivity analysis is a viable strategy.

Compliance with Ethical Standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Antolin P, Buffa A, Calabrò F, Martinelli M, Sangalli G (2015) Efficient matrix computation for tensor-product isogeometric analysis: the use of sum factorization. Comput Methods Appl Mech Eng 285:817–828
2. Antolin P, Buffa A, Coradello L (2020) A hierarchical approach to the a posteriori error estimation of isogeometric Kirchhoff plates and Kirchhoff–Love shells. Comput Methods Appl Mech Eng 363:112919
3. Arnout S, Firl M, Bletzinger KU (2012) Parameter free shape and thickness optimisation considering stress response. Struct Multidiscip Optim 45(6):801–814
4. Bandara K, Cirak F (2018) Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces. Comput Aided Des 95(3):62–71
5. Bischoff M, Ramm E, Irslinger J (2017) Encyclopedia of computational mechanics, vol 1859, 2nd edn. Wiley, Hoboken, pp 1–86
6. Blanchard L, Duvigneau R, Vuong AV, Simeon B (2014) Shape gradient for isogeometric structural design. J Optim Theory Appl 161(2):361–367
7. Bletzinger KU (2017) Shape optimization. American Cancer Society, Atlanta, pp 1–42
8. Bletzinger KU, Ramm E (2014) Computational form finding and optimization. In: Adriaenssens S, Block P, Veenendaal D, Williams C (eds) Shell structures for architecture. Routledge, London, pp 45–55
9. Bletzinger KU, Kimmich S, Ramm E (1991) Efficient modeling in shape optimal design. Comput Syst Eng 2(5–6):483–495
10. Bletzinger KU, Reitinger R, Kimmich S, Ramm E (1993) Shape optimization with program CARAT. Software systems for structural optimization, vol 110. Birkhäuser, Basel, pp 97–124
11. Bletzinger KU, Wüchner R, Daoud F, Camprubí N (2005) Computational methods for form finding and optimization of shells and membranes. Comput Methods Appl Mech Eng 194(30–33):3438–3452
12. Bletzinger KU, Firl M, Daoud F (2008) Approximation of derivatives in semi-analytical structural optimization. Comput Struct 86(13–14):1404–1416
13. Braibant V, Fleury C (1984) Shape optimal design using B-splines. Comput Methods Appl Mech Eng 44(3):247–267
14. Chamoin L, Thai H (2019) Certified real-time shape optimization using isogeometric analysis, PGD model reduction, and a posteriori error estimation. Int J Numer Methods Eng 119(3):151–176
15. Choi MJ, Cho S (2018) Constrained isogeometric design optimization of lattice structures on curved surfaces: computation of design velocity field. Struct Multidiscip Optim 58(1):17–34
16. Cohen E, Riesenfeld RF, Elber G (2001) Geometric modeling with splines: an introduction. A. K. Peters Ltd, Natick
17. Coniglio S, Gogu C, Amargier R, Morlier J (2019) Engine pylon topology optimization framework based on performance and stress criteria. AIAA J 57(12):5514–5526
18. Cottrell JA, Hughes TJR, Reali A (2007) Studies of refinement and continuity in isogeometric structural analysis. Comput Methods Appl Mech Eng 196(41–44):4160–4183
19. Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA, 1st edn. Wiley, Hoboken
20. Dang Manh N, Evgrafov A, Gravesen J, Lahaye D (2014) Iso-geometric shape optimization of magnetic density separators. COMPEL Int J Comput Math Electr Electron Eng 33(4):1416–1433
21. Daxini SD, Prajapati JM (2017) Parametric shape optimization techniques based on Meshless methods: a review. Struct Multidiscip Optim 56(5):1197–1214
22. Ding C, Cui X, Huang G, Li G, Tamma KK (2017) Exact and efficient isogeometric reanalysis of accurate shape and boundary modifications. Comput Methods Appl Mech Eng 318:619–635
23. Ding C et al (2018) A gradient-based shape optimization scheme via isogeometric exact reanalysis. Eng Comput. 35(8):2696–2721. https://doi.org/10.1108/EC-08-2017-0292
24. Duysinx P, Sigmund O (1998) New developments in handling stress constraints in optimal material distribution. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. American Institute of Aeronautics and Astronautics, Reston, Virigina, pp 1501–1509
25. Echter R (2013) Isogeometric analysis of shells. PhD thesis, Universität Stuttgart
26. Echter R, Oesterle B, Bischoff M (2013) A hierarchic family of isogeometric shell finite elements. Comput Methods Appl Mech Eng 254:170–180
27. Ertl FJ, Dhondt G, Bletzinger KU (2019) Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures. Comput Methods Appl Mech Eng 353:86–106
28. Evans E, Scott M, Li X, Thomas D (2015) Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. Comput Methods Appl Mech Eng 284:1–20
29. Evans JA, Bazilevs Y, Babuška I, Hughes TJ (2009) n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. Comput Methods Appl Mech Eng 198(21–26):1726–1741
30. Farin G (2002) Curves and surfaces for CAGD: a practical guide, 5th edn. Morgan Kaufmann Publishers Inc., San Francisco
31. Firl M (2010) Optimal shape design of shell structures. PhD thesis, Technische Universität München
32. Fußeder D, Simeon B, Vuong AV (2015) Fundamental aspects of shape optimization in the context of isogeometric analysis. Comput Methods Appl Mech Eng 286:313–331. https://doi.org/10.1016/j.cma.2014.12.028
33. Gígola C, Gomez S (1990) A regularization method for solving the finite convex min-max problem. SIAM J Numer Anal 27(6):1621–1634. https://doi.org/10.1137/0727095
34. Gravesen J, Evgrafov A, Nguyen DM (2011) On the sensitivities of multiple eigenvalues. Struct Multidiscip Optim 44(4):583–587. https://doi.org/10.1007/s00158-011-0644-9
35. Haftka RT, Grandhi RV (1986) Structural shape optimization—a survey. Comput Methods Appl Mech Eng 57(1):91–106. https://doi.org/10.1016/0045-7825(86)90072-1
36. Hassani B, Tavakkoli S, Moghadam N (2011) Application of isogeometric analysis in structural shape optimization. Sci Iran 18(4):846–852. https://doi.org/10.1016/j.scient.2011.07.014
37. Hiemstra RR, Sangalli G, Tani M, Calabrò F, Hughes TJ (2019) Fast formation and assembly of finite element matrices with application to isogeometric linear elasticity. Comput Methods Appl Mech Eng 355:234–260. https://doi.org/10.1016/j.cma.2019.06.020
38. Hirschler T, Bouclier R, Duval A, Elguedj T, Morlier J (2019a) Isogeometric sizing and shape optimization of thin structures with a solid-shell approach. Struct Multidiscip Optim 59(3):767–785. https://doi.org/10.1007/s00158-018-2100-6
39. Hirschler T, Bouclier R, Duval A, Elguedj T, Morlier J (2019b) The embedded isogeometric Kirchhoff–Love shell: from design to shape optimization of non-conforming stiffened multipatch structures. Comput Methods Appl Mech Eng 349:774–797

40. Hojjat M, Stavropoulou E, Bletzinger KU (2014) The vertex morphing method for node-based shape optimization. Comput Methods Appl Mech Eng 268:494–513
41. Hsu YL (1994) A review of structural shape optimization. Comput Ind 25(1):3–13
42. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194(39–41):4135–4195
43. Imam MH (1982) Three-dimensional shape optimization. Int J Numer Methods Eng 18(5):661–673
44. Johnson SG. The NLopt nonlinear-optimization package. http://github.com/stevengj/nlopt
45. Kamensky D et al (2017) Immersogeometric cardiovascular fluid-structure interaction analysis with divergence-conforming B-splines. Comput Methods Appl Mech Eng 314:408–472
46. Kang P, Youn SK (2016) Isogeometric shape optimization of trimmed shell structures. Struct Multidiscip Optim 53(4):825–845
47. Kegl M, Brank B (2006) Shape optimization of truss-stiffened shell structures with variable thickness. Comput Methods Appl Mech Eng 195(19–22):2611–2634
48. Kiendl J (2011) Isogeometric analysis and shape optimal design of shell structures. PhD thesis, Technische Universität München
49. Kiendl J, Bletzinger KU, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. Comput Methods Appl Mech Eng 198(49–52):3902–3914
50. Kiendl J, Schmidt R, Wuchner R, Bletzinger KU (2014) Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput Methods Appl Mech Eng 274:148–167
51. Kirchhoff G (1850) Über das gleichgewicht und die bewegung einer elastischen scheibe. Journal für die reine und angewandte Mathematik 40:51–88
52. Kirsch U (2003) A unified reanalysis approach for structural analysis, design, and optimization. Struct Multidiscip Optim 25(2):67–85
53. Kraft D (1988) A software package for sequential quadratic programming. Forschungsbericht, Wiss. Berichtswesen d. DFVLR, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln
54. Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 200(9–12):985–996
55. Lee BG, Park Y (2002) Degree elevation of nurbs curves by weighted blossom. Korean J Comput Appl Math 9(1):151–165
56. Lei Z, Gillot F, Jezequel L (2018) Shape optimization for natural frequency with isogeometric Kirchhoff–Love shell and sensitivity mapping. Math Probl Eng 2018:1–11
57. Li K, Qian X (2011) Isogeometric analysis and shape optimization via boundary integral. Comput Aided Des 43(11):1427–1437
58. Li XS, Fang SC (1997) On the entropic regularization method for solving min-max problems with applications. Math Methods Oper Res 46(1):119–130
59. Lian H, Kerfriden P, Bordas S (2017) Shape optimization directly from CAD: an isogeometric boundary element approach using T-splines. Comput Methods Appl Mech Eng 317:1–41
60. Lipton S, Evans J, Bazilevs Y, Elguedj T, Hughes T (2010) Robustness of isogeometric structural discretizations under severe mesh distortion. Comput Methods Appl Mech Eng 199(5–8):357–373
61. Love AEH (1888) Xvi. The small free vibrations and deformation of a thin elastic shell. Philos Trans R Soc Lond A 179:491–546
62. Lund E, Stegmann J (2006) Eigenfrequency and buckling optimization of laminated composite shell structures using discrete material optimization. In: IUTAM symposium on topological design optimization of structures, machines and materials, vol 137. Springer, Netherlands, pp 147–156
63. Manh ND, Evgrafov A, Gersborg AR, Gravesen J (2011) Isogeometric shape optimization of vibrating membranes. Comput Methods Appl Mech Eng 200(13–16):1343–1353
64. Morganti S et al (2015) Patient-specific isogeometric structural analysis of aortic valve closure. Comput Methods Appl Mech Eng 284:508–520
65. Nagy AP, Abdalla MM, Gürdal Z (2010) Isogeometric sizing and shape optimisation of beam structures. Comput Methods Appl Mech Eng 199(17–20):1216–1230
66. Nagy AP, Abdalla MM, Gürdal Z (2011) Isogeometric design of elastic arches for maximum fundamental frequency. Struct Multidiscip Optim 43(1):135–149
67. Nagy AP, IJsselmuiden ST, Abdalla MM (2013) Isogeometric design of anisotropic shells: optimal form and material distribution. Comput Methods Appl Mech Eng 264:145–162
68. Nguyen DM, Evgrafov A, Gravesen J (2012) Isogeometric shape optimization for electromagnetic scattering problems. Prog Electromagn Res B 45:117–146
69. Occelli M, Elguedj T, Bouabdallah S, Morançay L (2019) LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis. Adv Eng Softw 131:166–185
70. Olhoff N (1989) Multicriterion structural optimization via bound formulation and mathematical programming. Struct Optim 1(1):11–17
71. Olhoff N, Rasmussen J, Lund E (1993) A method of “exact” numerical differentiation for error elimination in finite-element-based semi-analytical shape sensitivity analyses*. Mech Struct Mach 21(1):1–66
72. Papadrakakis M, Lagaros ND, Tsompanakis Y, Plevris V (2001) Large scale structural optimization: computational methods and optimization algorithms. Arch Comput Methods Eng 8(3):239–301
73. Park BU, Seo YD, Sigmund O, Youn SK (2013) Shape optimization of the stokes flow problem based on isogeometric analysis. Struct Multidiscip Optim 48(5):965–977
74. Piegl L, Tiller W (1997) The NURBS book, 2nd edn. Springer, Berlin
75. Qian X (2010) Full analytical sensitivities in NURBS based isogeometric shape optimization. Comput Methods Appl Mech Eng 199(29–32):2059–2071
76. Seyranian AP, Lund E, Olhoff N (1994) Multiple eigenvalues in structural optimization problems. Struct Optim 8(4):207–227
77. Shamanskiy A, Gfrerer MH, Hinz J, Simeon B (2020) Isogeometric parametrization inspired by large elastic deformation. Comput Methods Appl Mech Eng 363:112920
78. Shimoda M, Liu Y (2014) A non-parametric free-form optimization method for shell structures. Struct Multidiscip Optim 50(3):409–423
79. Shimoda M, Liu Y (2016) Node-based free-form optimization method for vibration problems of shell structures. Comput Struct 177:91–102
80. Shimoda M, Nagano T, Morimoto T, Liu Y, Shi JX (2016) Non-parametric free-form optimal design of frame structures in natural frequency problem. Int J Mech Sci 117:334–345
81. Sigmund O (1997) On the design of compliant mechanisms using topology optimization*. Mech Struct Mach 25(4):493–524
82. Stanford B, Beran P, Bhatia M (2014) Aeroelastic topology optimization of blade-stiffened panels. J Aircr 51(3):938–944
83. Taheri A, Hassani B (2014) Simultaneous isogeometrical shape and material design of functionally graded structures for optimal eigenfrequencies. Comput Methods Appl Mech Eng 277:46–80
84. Tomlow J (2011) Gaudí’s reluctant attitude towards the inverted catenary. Proc Inst Civ Eng Eng Hist Herit 164(4):219–233

85. Van Keulen F, Haftka R, Kim N (2005) Review of options for structural design sensitivity analysis. Part 1: linear systems. Comput Methods Appl Mech Eng 194(30–33):3213–3243
86. Verbart A, Langelaar M, van Keulen F (2017) A unified aggregation and relaxation approach for stress-constrained topology optimization. Struct Multidiscip Optim 55(2):663–679
87. Wall WA, Frenzel MA, Cyron C (2008) Isogeometric structural shape optimization. Comput Methods Appl Mech Eng 197(33–40):2976–2988
88. Wang W, Clausen PM, Bletzinger KU (2015) Improved semi-analytical sensitivity analysis using a secant stiffness matrix for geometric nonlinear shape optimization. Comput Struct 146:143–151
89. Wang Y, Wang Z, Xia Z, Hien Poh L (2018) Structural design optimization using isogeometric analysis: a comprehensive review. Comput Model Eng Sci 117(3):455–507
90. Wang ZP, Turteltaub S (2015) Isogeometric shape optimization for quasi-static processes. Int J Numer Methods Eng 104(5):347–371
91. Wang ZP, Abdalla M, Turteltaub S (2017a) Normalization approaches for the descent search direction in isogeometric shape optimization. Comput Aided Des 82:68–78
92. Wang ZP, Turteltaub S, Abdalla M (2017b) Shape optimization and optimal control for transient heat conduction problems using an isogeometric approach. Comput Struct 185:59–74
93. Weeger O, Narayanan B, Dunn ML (2018) Isogeometric shape optimization of nonlinear, curved 3D beams and beam structures. Comput Methods Appl Mech Eng 345:26–51. https://doi.org/10.1016/j.cma.2018.10.038
94. Williams CJK (2014) What is a shell? In: Adriaenssens S, Block P, Veenendaal D, Williams C (eds) Shell structures for architecture. Routledge, London, pp 21–31
95. Xia Q, Shi T, Wang MY (2011) A level set based shape and topology optimization method for maximizing the simple or repeated first eigenvalue of structure vibration. Struct Multidiscip Optim 43(4):473–485
96. Xu M, Wang S, Xie X (2019) Level set-based isogeometric topology optimization for maximizing fundamental eigenfrequency. Front Mech Eng 14(2):222–234
97. Yang RJ, Chen CJ (1996) Stress-based topology optimization. Struct Optim 12(2–3):98–105
98. Zhu B et al (2020) Design of compliant mechanisms using continuum topology optimization: a review. Mech Mach Theory 143:103622