Structural and Multidisciplinary Optimization (2024) 67:124
https://doi.org/10.1007/s00158-024-03843-6

RESEARCH PAPER

Robust design optimization using a non-intrusive second-order approximation of stochastic moments

Jan Christoph Krüger[1] · **Benedikt Kriegesmann**[1]

Received: 22 February 2024 / Revised: 25 June 2024 / Accepted: 28 June 2024 / Published online: 10 July 2024

Abstract
This paper presents a new formulation of the second-order fourth-moment method (sometimes referred to as second-order perturbation method or second-order method of moments). The method allows to efficiently predict the stochastic moments of a response function and is therefore often used within robust design optimization. The new approach allows a non-intrusive implementation at the same cost as existing, highly intrusive formulations. Therefore, the new approach can be applied to any objective function without significant implementation effort. It is based on a few finite difference steps into special directions and hence is dependent on the corresponding step sizes. An automatic step size procedure is supplied beside a detailed convergence analysis. The advantages of the new formulation are demonstrated by robust design optimizations of a 2D and a 3D example using the geometrically nonlinear finite element method.

Keywords Robust design optimization (RDO) · Robust topology optimization · Method of moments · Second-order perturbation approach

1 Introduction

Structural optimization became a popular field in the domain of structural mechanics since it helps to find highly efficient designs for a given purpose. In combination with the adjoint method, gradient-based optimization schemes lead to good designs with only a few finite element simulations (Sigmund 2011). However, it is known that design optimization might lead to designs that are sensitive to imperfections. In consequence, many approaches for optimization under uncertainty have been presented during the last decades. According to Yao et al. (2011) and Kanno (2020) the major types of optimization under uncertainty are reliability-based design optimization [e.g. (Schuëller and Valdebenito 2010)], worst-case optimization [e.g. (Henrichsen et al. 2015; Hederberg and Thore 2023)] and robust design optimization [e.g. (Schevenels et al. 2011)]. In addition, a distinction between aleatory and epistemic uncertainty is possible (Helton et al. 2004). While epistemic uncertainty is quantified using possibility theory [e.g fuzzy arithmetic (MäMäck et al. 2019; Papaioannou et al. 2019)], aleatory uncertainty is tackled using probabilistic methods.

The current contribution is located in the field of robust design optimization (RDO) using probabilistic methods. In RDO, mean and standard deviation of an objective are minimized. There are several approaches to compute these stochastic quantities. For instance, Schevenels et al. applied the Monte-Carlo method to topology optimization in Schevenels et al. (2011). This method is easy to implement and very accurate if a sufficient number of samples is considered. However, the Monte Carlo method requires one FE-simulation per sample leading to extremely high computational cost.

There are several approaches to reduce the cost of the Monte Carlo method based on surrogate models (Chatterjee et al. 2019), such as neural networks or the non-intrusive polynomial chaos expansion (PCE). In the PCE, the system response is approximated using Legendre-polynomials (Keshavarzzadeh et al. 2017). The coefficients of the polynomial are computed using Gauß-Legendre integration. Using efficient sampling methods, the number of function evaluations approximately scales linearly with the number of random

Responsible editor: Zhen Hu

✉ Jan Christoph Krüger
jan.krueger@tuhh.de

[1] Institute for Structural Mechanics in Lightweight Design, Hamburg University of Technology, Eißendorfer Straße 40, 21073 Hamburg, Germany

parameters for a linear expansion and quadratically for a quadratic expansion. The training of the surrogate model must be done in every optimization iteration leading to huge computational cost for many random parameters. Hence, different approaches have been proposed to reduce the number of random parameters such as reduced order models (Torres et al. 2021).

Another group of approaches is based on a Taylor series expansion of the system response. The perturbation approach is based on a Taylor series of the deformation vector. The stochastic moments of the objective are computed analytically from the approximated eformation covariance matrix. covariance matrix. This approach has been widely applied to truss structures [e.g. (Doltsinis et al. 2005; Asadpoure et al. 2011; Changizi and Jalalpour 2017; Valm et al. 2022)] and to topology optimization [e.g. (Lazarov et al. 2012)]. Similar to the non-intrusive polynomial chaos expansion, the number of systems of equations scales linearly with the number of random parameters for a first-order approximation and quadratically for a second-order approximation.

As an alternative, the objective function itself can be approximated by a Taylor series, leading to the first-order second-moment method in the case of a first-order approximation and to the second-order forth-moment method in the case of a second-order approximation. In Kranz et al. (2023), a generalized formulation of the first-order second-moment method is derived, where only two additional systems of equations have to be solved independently of the number of random parameters. The authors derived a non-intrusive version of the first-order second-moment method in Krüger et al. (2023), which only requires one additional function evaluation at all. However, first-order approximations are very inaccurate in many situations (Kriegesmann 2021) and therefore often cannot be used for robust design optimization.

As an alternative, a second-order approach can be utilized, which is referred to as "mean-variance approach" (Beyer and Sendhoff 2007), "second-order method of moments" (Green et al. 2002) or "second-order fourth moment method (SOFM)" (Kriegesmann et al. 2011). Jansen et al. applied this method to robust topology optimization for non-linear compliance (Jansen et al. 2015) using the adjoint method for gradient computation. In the mentioned publication, the method is called "second-order perturbation method". Using the proposed approach given by Jansen et al. the number of systems of equation scales linear with the number of random variables and hence is much more effective than the aforementioned approaches. However, this method is highly intrusive and requires storing large matrices, which limits the scalability of the approach. A similar efficiency is achieved by Tan et al. in Tan and Faghihi (2024), but again the resulting method is highly intrusive.

The current publication presents a non-intrusive version of the second-order fourth-moment method which has a comparable computational cost to the approach presented in Jansen et al. (2015). It is based on similar ideas as the principal sensitivity first-order second-moment method presented in Krüger et al. (2023). Due to its non-intrusive nature, the memory requirements are very low and it can be applied for any kind of optimization problem such as topology optimization or shape optimization.

The paper is organized as follows: First, the second-order fourth-moment method is derived and simplified for the special case of normal distributed random parameters. Next, a standard finite difference framework is presented and analysed. Afterwards, the new approach called principal sensitivity second-order fourth-moment is derived. The new approach is analysed in terms of stability and performance. Afterwards it is applied to a shape optimization of a 3D compression bar, the 2D shallow arch topology optimization example given in Jansen et al. (2015) as well as a 3D shallow arch example with around 3.5 million finite elements.

2 Robust design optimization using the second-order fourth-moment method

In robust design optimization, the structure of the optimization problem is similar to a deterministic optimization. The whole parametrization (e.g. density-based topology optimization parametrization, size optimization parametrization) does not change. However, the objective (or constraint) differs significantly since a robust objective (or constraint) function given by

$$f_p(\boldsymbol{x},\boldsymbol{y}) = \mu_f(\boldsymbol{x},\boldsymbol{y}) + \kappa\sigma_f(\boldsymbol{x},\boldsymbol{y}) \tag{1}$$

is used. Here $\boldsymbol{y}$ represents the vector of design variables and $\boldsymbol{x}$ represents the vector of random variables. The mean value μ_f and the standard deviation σ_f of the objective function f are combined using a weighting factor κ. In order to compute the mean and the standard deviation, a probabilistic method (e.g. the second-order fourth-moment method) is embedded into the basic optimization framework. The weighting factor can be chosen arbitrarily. Under the assumption that the objective function follows a known distribution, the weighting factor indicates the probability that a sample behaves worse than the robust objective value.

In this contribution, a new framework based on the second-order fourth-moment method is presented. The basic equations have been derived in Doltsinis et al. (2005), Asadpoure et al. (2011) in the context of the perturbation method and in Jansen et al. (2015), Kriegesmann et al. (2011). The basic concept of the second-order fourth-moment method is

to approximate the objective function using a Taylor series T given by

$$f(\boldsymbol{x}) \approx T(\boldsymbol{x}) = f_0 + \sum_{i=1}^{M} \frac{df}{dx_i} \cdot (x_i - \mu_i) + \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{d^2 f}{dx_i dx_j} \cdot (x_i - \mu_i)(x_j - \mu_j) \tag{2}$$

with the objective value in the mean case f_0 and the mean value μ_i of x_i (Kriegesmann et al. 2011). Analytic equations for mean and variance are given by

$$\mu_f = \int_{-\infty}^{\infty} f(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{3}$$

$$\sigma_f^2 = \int_{-\infty}^{\infty} (f(\boldsymbol{x}) - \mu_f)^2 p(\boldsymbol{x}) d\boldsymbol{x} \tag{4}$$

with the probability density function $p(\boldsymbol{x})$ (Haldar and Mahadevan 1999). The integral is applied into all dimensions of $\boldsymbol{x}$. Inserting the Taylor approximation (2) to (3) and (4) leads to the approximations

$$\mu_f \approx f_0 + \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{d^2 f}{dx_i dx_j} cov(x_i, x_j) \tag{5}$$

$$\begin{aligned} \sigma_f^2 \approx & \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{df}{dx_i} \frac{df}{dx_j} cov(x_i, x_j) \\ & - \frac{1}{4} \sum_{i=1}^{M} \sum_{j=1}^{M} \left(\frac{d^2 f}{dx_i dx_j} \right)^2 cov(x_i, x_j)^2 \\ & + \frac{1}{4} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{k=1}^{M} \sum_{l=1}^{M} \frac{d^2 f}{dx_i dx_j} \frac{d^2 f}{dx_k dx_l} \mu_{i,j,k,l} \\ & + \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{k=1}^{M} \frac{d^2 f}{dx_i dx_j} \frac{df}{dx_k} \mu_{i,j,k} \quad , \end{aligned} \tag{6}$$

where $cov(x_i, x_j)$ denotes the covariance between the variables x_i and x_j. The operators $\mu_{i,j,k}$ and $\mu_{i,j,k,l}$ represent the third and the fourth central stochastic moments between the variables x_i, x_j, x_k, x_l. The standard deviation is computed from the variance. Under the assumption of normal distributed random variables, the third stochastic moment vanishes and Isserlis theorem (Isserlis 1918) can be applied to compute the fourth order stochastic moments using the covariance. Hence, Eq. (6) simplifies to

$$\begin{aligned} \sigma_f^2 = & \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{df}{dx_i} \cdot \frac{df}{dx_j} \cdot cov(x_i, x_j) \\ & + \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{k=1}^{M} \sum_{l=1}^{M} \frac{d^2 f}{dx_i dx_j} \frac{d^2 f}{dx_k dx_l} \cdot cov(x_i, x_k) cov(x_j, x_l) \end{aligned} \quad . \tag{7}$$

If the random variables are also uncorrelated, (5) and (7) further simplify to

$$\mu_f = f_0 + \frac{1}{2} \sum_{i=1}^{M} \frac{d^2 f}{dx_i^2} \sigma_{xi}^2 \tag{8}$$

$$\sigma_f^2 = \sum_{i=1}^{M} \left(\frac{df}{dx_i} \right)^2 \cdot \sigma_{xi}^2 + \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left(\frac{d^2 f}{dx_i dx_j} \right)^2 \cdot \sigma_{xi}^2 \sigma_{xj}^2 \quad . \tag{9}$$

Here σ_{xi}^2 denotes the variance of the random variable x_i.

In the following, the simple case of uncorrelated random variables which follow a normal distribution is considered. This is only a small limitation since variables that do not follow a normal distribution can be transformed to normal distributed variables using a translation process (Grigoriu Mircea 1998). In addition, correlated random variables can be transformed to uncorrelated variables (Kiureghian 2022).

In many design optimization disciplines, such as topology optimization, only gradient-based methods lead to good designs in appropriate time (Sigmund 2011). Hence, the gradients of mean and standard deviation must be computed. Using direct differentiation, the gradients read

$$\frac{d\mu_f}{d\boldsymbol{y}} = \frac{df}{d\boldsymbol{y}} + \frac{1}{2} \sum_{i=1}^{M} \frac{d^3 f}{dx_i^2 d\boldsymbol{y}} \sigma_{xi}^2 \tag{10}$$

$$\begin{aligned} \frac{d\sigma_f^2}{d\boldsymbol{y}} = & 2 \sum_{i=1}^{M} \frac{d^2 f}{dx_i d\boldsymbol{y}} \frac{df}{dx_i} \cdot \sigma_{xi}^2 \\ & + \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{d^3 f}{dx_i dx_j d\boldsymbol{y}} \frac{d^2 f}{dx_i dx_j} \cdot \sigma_{xi}^2 \sigma_{xj}^2 \quad . \end{aligned} \tag{11}$$

The gradient of the standard deviation is computed from the variance by

$$\frac{d\sigma_f}{d\boldsymbol{y}} = \frac{1}{2\sigma_f} \frac{d\sigma_f^2}{d\boldsymbol{y}} \quad . \tag{12}$$

3 Existing approaches

In practical application there are several approaches to solve the Eqs. (8), (9), (10) and (11). In general, a distinction between analytic/exact and finite difference frameworks is possible. While finite difference frameworks use the objective as a black box, analytic methods require further information.

3.1 Analytic approaches

A naive approach would be to evaluate the governing equations of the second-order fourth-moment method directly. In this case, the derivatives $\frac{df}{d\boldsymbol{y}}$, $\frac{df}{dx_i}$, $\frac{d^2f}{dx_i dx_j}$, $\frac{d^3f}{dx_i dx_j d\boldsymbol{y}}$ are computed analytically and afterwards inserted into (8), (9), (10) and (11). The computation of the derivatives can be done with any method such as direct differentiation, the adjoint method or automatic differentiation. In Kriegesmann and Lüdeker (2019) this approach is applied to an example with random loads. There, all derivatives are computed using the adjoint method. It is shown that computing the third order derivatives requires the solution of several adjoint systems. Depending on the formulation, the number of adjoint systems scales bilinear with the size of $\boldsymbol{y}$ and $\boldsymbol{x}$ or it scales quadratically with the size of $\boldsymbol{x}$. Hence the authors conclude that this approach can only be applied to very small systems.

A more efficient approach is given in Jansen et al. (2015) for a general nonlinear finite element formulation. Here, the adjoint method is applied to the robust objective itself to find the gradient. A detailed derivation can be found in B. First, the robust objective is evaluated using $\frac{df}{dx_i}$ and $\frac{d^2f}{dx_i dx_j}$. For that, several (adjoint) systems have to be solved. Afterwards, the robust objective function is augmented using the previously used (adjoint) systems. Differentiation and reordering leads to an explicit equation for the gradient of the robust objective and several new adjoint systems. The number of adjoint systems scales linearly with the size of $\boldsymbol{x}$. Hence, this approach is much more effective than the framework given in Kriegesmann and Lüdeker (2019).

Both methods are analytically derived and hence exact. The computation of the derivatives of the objective requires the solution of many adjoint systems. However, the system matrix stays constant while only the right hand side changes. In consequence, both approaches can be strongly accelerated using a matrix decomposition (e.g. Cholesky decomposition) of the system matrix. However this is only possible for rather small systems because the memory requirements of storing an exact matrix decomposition becomes infeasibly high, even for moderate models.

Apart from the advantages (especially the approach given by Jansen et al.) the aforementioned frameworks have significant disadvantages. First, the memory requirements are often very high. Both methods rely on several partial derivatives, such as $\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{u}}$ where $\boldsymbol{K}_T$ represents the tangential stiffness matrix and $\boldsymbol{u}$ the deformation vector. These partial derivatives are often very memory consuming compared to the system matrix. Furthermore, the aforementioned approaches are fully problem dependent. This means that all derivations must be done for every combination of objective, design variables and random variables. Especially higher-order derivatives have coupled terms leading to a large number of possible combinations. This results in a high effort for implementing a new problem. In addition, the user might not have full access to the source code of the used software. This holds especially for commercial code. While many tools (such as Abaqus, Ansys) deliver first order derivatives for many combinations of objective function and parameters, higher-order derivatives are usually not given. Finally, there are objective functions (e.g. eigenfrequency) which provide a singular coefficient matrix of the linear equation system and adjoint systems with non-zero right-hand sides cannot be solved (Krüger and Kriegesmann 2024).

3.2 Finite difference framework

In difference to the analytic approaches, the finite difference approach is a black box method. Hence, it is non-intrusive and applicable to commercial code. It is assumed that the function value f and the gradient with respect to design variables $\frac{df}{d\boldsymbol{y}}$ are delivered by one function evaluation. The remaining required derivatives $\frac{df}{dx_i}$, $\frac{d^2f}{dx_i dx_j}$, $\frac{d^3f}{dx_i dx_j d\boldsymbol{y}}$ are computed using finite differences and are afterwards inserted into (8), (9), (10) and (11). A detailed derivation of the used finite difference approximations including error estimation is given in A.

The mean of the objective is computed using the function value f itself and all diagonal entries of the hessian $\frac{d^2f}{dx_i dx_i}$ with respect to the random parameters. The most effective finite difference framework for computing homogeneous second-order sensitivities are central differences given by

$$\frac{d^2f}{dx_i^2} \approx \frac{f(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i) - 2f(\mu_{\boldsymbol{x}}) + f(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i)}{\Delta x_i^2} \tag{13}$$

with an error of $\mathcal{O}(\Delta x_i^2)$ at the cost of two function evaluations per random variable. Here, $\boldsymbol{\Delta x}_i$ denotes a perturbation step into the i-th direction with the length Δx_i. The gradient with respect to design variables is computed by applying central differences to the gradient of the objective leading to

$$\frac{d^3 f}{dx_i^2 d\boldsymbol{y}} \approx \frac{\frac{df(\mu_x+\Delta \boldsymbol{x}_i)}{d\boldsymbol{y}} - 2\frac{df(\mu_x)}{d\boldsymbol{y}} + \frac{df(\mu_x-\Delta \boldsymbol{x}_i)}{d\boldsymbol{y}}}{\Delta x_i^2} \quad . \tag{14}$$

Again a convergence rate of $\mathcal{O}(\Delta x_i^2)$ is achieved at the cost of two gradient evaluations per random variable.

Computing the variance of the robust objective requires the gradient $\frac{df}{dx_i}$ as well as the full hessian matrix $\frac{d^2 f}{dx_i dx_j}$. Since $f(x_i + \Delta x_i)$ and $f(x_i - \Delta x_i)$ are already computed, $\frac{df}{dx_i}$ is calculated using central differences given by

$$\frac{df}{dx_i} \approx \frac{f(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_i) - f(\mu_{\boldsymbol{x}} - \Delta \boldsymbol{x}_i)}{2\Delta x_i} \tag{15}$$

without extra cost and an error of $\mathcal{O}(\Delta x_i^2)$. (Here, also forward finite differences could be applied. However, that would reduce the convergence rate without any benefit.) The diagonal elements of the hessian have already been computed. Hence, only the non-diagonal elements have to be computed additionally. Further reduction of computational effort is achieved by only considering the lower diagonal elements since the hessian is always symmetric. Using (A.9), the mixed derivatives read as

$$\frac{d^2 f}{dx_i dx_j} \approx \frac{f(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_i + \Delta \boldsymbol{x}_j) + f(\mu_{\boldsymbol{x}}) - f(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_i) - f(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_j)}{\Delta x_i \Delta x_j} \tag{16}$$

with an error of $\mathcal{O}(\Delta x_i) + \mathcal{O}(\Delta x_j)$. The number of additional function evaluations is given by $\frac{M^2}{2} - \frac{M}{2}$ where M is the number of random variables. Similar to the mean, the gradients with respect to design variables are computed by applying the finite difference equations to the gradient of the objective. Hence the quantities read as

$$\frac{d^2 f}{dx_i d\boldsymbol{y}} \approx \frac{\frac{df(\mu_x+\Delta \boldsymbol{x}_i)}{d\boldsymbol{y}} - \frac{df(\mu_x-\Delta \boldsymbol{x}_i)}{d\boldsymbol{y}}}{2\Delta x_i} \tag{17}$$

$$\frac{d^3 f}{dx_i dx_j d\boldsymbol{y}} \approx \frac{\frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_i + \Delta \boldsymbol{x}_j) + \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}}) - \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_i) - \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \Delta \boldsymbol{x}_j)}{\Delta x_i \Delta x_j} \quad . \tag{18}$$

Over all, the number of function evaluations is given by $\frac{M^2}{2} + 1.5M + 1$, where M is the number of random variables. The overall convergence rate is dominated by $\mathcal{O}(\Delta x_i) + \mathcal{O}(\Delta x_j)$. The errors of the gradients are consistent to the related functions, since the same finite difference frameworks are applied to function and gradient. Hence small numeric errors should not disturb the optimization procedure. If a convergence rate of $\mathcal{O}(\Delta x_i^2) + \mathcal{O}(\Delta x_j^2)$ should be achieved, (A.11) must be applied. In consequence the computational cost increases to $2M^2 + 1$ function evaluations.

The finite difference framework is fully non-intrusive. Hence, it can be applied to any commercial tool without a significant effort, similar to the implementation of the first-order second-moment method presented by Steltner et al. (2022). At the same time, it can be fully parallelized since all steps are known a-priori. If theoretically a sufficient number of CPU-cores and memory are available, the computation time is equal to the computation time of a deterministic optimization. However, the required amount of resources would be extreme already for a moderate number of random variables.

4 Proposed approach—principal sensitivity second-order fourth-moment method

The previous approaches either use much information (e.g. stiffness derivatives, partial objective derivatives) in order to gain high efficiency or only use black-box information with a low accuracy. The required information for the analytic methods can only be gained with high implementation effort if it is accessible at all.

An alternative new non-intrusive framework is presented which additionally uses the gradient with respect to the random variables $\frac{df}{d\boldsymbol{x}}$ as an input to improve efficiency. It shows similarities to the principal sensitivity first-order second-moment method (PSFOSM) presented in Krüger et al. (2023) and hence, it is called principal sensitivity second-order fourth-moment method (PSSOFM).

There are many ways to get the required gradients, even using commercial tools. In the case of in-house optimization tools, the gradient for a new combination of objective and parameters can be implemented without high effort. Many partial derivatives and adjoint systems do not differ from the computation of $\frac{df}{d\boldsymbol{y}}$, which is already needed for deterministic optimization. Hence, only very few partial derivatives have to be added. Since first-order sensitivities have no couplings, different sets of variables can be merged without additional effort. If commercial tools are used, many gradients are implemented for optimization purpose. Gradients of various objectives used for topology optimization, shape optimization, bead optimization and sizing contain derivatives with respect to density/stiffness, node coordinates and feature

sizes. These derivatives can be regarded as (or mapped to) derivatives with respect to random parameters. In addition, it is possible to implement own sensitivities using the software interfaces as in Sartorti et al. (2023).

Using the same central differences like in the aforementioned finite difference framework leads to an error of order $\mathcal{O}(\Delta x_i^2)$ where the number of function evaluations scales linear with the number of random variables. The authors do not see any possible (non-intrusive) improvement and hence propose to use central differences (13) and (14) for (8) and (10) leading to

$$\mu_f \approx f_0 + \frac{1}{2}\sum_{i=1}^{M} \frac{f(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i) - 2f(\mu_{\boldsymbol{x}}) + f(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i)}{\Delta x_i^2}\sigma_{xi}^2 \tag{19}$$

$$\frac{d\mu_f}{d\boldsymbol{y}} \approx \frac{df}{d\boldsymbol{y}} + \frac{1}{2}\sum_{i=1}^{M} \frac{\frac{df(\mu_{\boldsymbol{x}}+\boldsymbol{\Delta x}_i)}{d\boldsymbol{y}} - 2\frac{df(\mu_{\boldsymbol{x}})}{d\boldsymbol{y}} + \frac{df(\mu_{\boldsymbol{x}}-\boldsymbol{\Delta x}_i)}{d\boldsymbol{y}}}{\Delta x_i^2}\sigma_{xi}^2 \quad , \tag{20}$$

where M represents the number of random variables.

The second-order approximation of the variance (9) requires not only second-order derivatives with respect to one variable x_i, but also mixed partial derivatives. In order to circumvent using (16), central differences are applied to $\frac{df}{d\boldsymbol{x}}$

$$\frac{d^2f}{d\boldsymbol{x}dx_i} \approx \frac{\frac{df}{d\boldsymbol{x}}(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i) - \frac{df}{d\boldsymbol{x}}(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i)}{2\Delta x_i} \quad , \tag{21}$$

again with a convergence rate $\mathcal{O}(\Delta x_i^2)$. This operation does not produce significant additional cost because the problem is already solved at the same evaluation points for Eq. (19).

The highest order derivatives occur in the approximation of the gradient of the variance (11), where $\frac{d^2f}{dx_i d\boldsymbol{y}}$ and $\frac{d^3f}{dx_i dx_j d\boldsymbol{y}}$ have to be computed. Like in the finite difference framework, $\frac{d^2f}{dx_i d\boldsymbol{y}}$ is computed using (17) with an error of order $\mathcal{O}(\Delta x_i^2)$. Again this operation does not produce significant computational cost because the problem is already solved at the same evaluation points for Eq. (19). Instead of computing $\frac{d^3f}{dx_i dx_j d\boldsymbol{y}}$ directly, the second part of (11) is reformulated as

$$\sum_{i=1}^{M}\sum_{j=1}^{M} \frac{d^3f}{dx_i dx_j d\boldsymbol{y}} \frac{d^2f}{dx_i dx_j} \cdot \sigma_{xi}^2\sigma_{xj}^2 = \sum_{i}^{M} \frac{d^3f}{dx_i d\boldsymbol{y} d\boldsymbol{x}} \boldsymbol{Cov} \frac{d^2f}{dx_i d\boldsymbol{x}}\sigma_{xi}^2 \tag{22}$$

with the (diagonal) covariance matrix $\boldsymbol{Cov}$. Further inspection shows that $\frac{d^3f}{dx_i d\boldsymbol{y} d\boldsymbol{x}}$ is a matrix and $\boldsymbol{Cov}\frac{d^2f}{dx_i d\boldsymbol{x}}\sigma_{xi}^2$ is a vector. This matrix vector product is interpreted as a directional derivative, which is different for every entry of x_i. Hence it holds

$$\sum_{i=1}^{M}\sum_{j=1}^{M} \frac{d^3f}{dx_i dx_j d\boldsymbol{y}} \frac{d^2f}{dx_i dx_j} \cdot \sigma_{xi}^2\sigma_{xj}^2 = \sum_{i}^{M} \frac{d^3f}{dx_i d\boldsymbol{y} ds_i} \tag{23}$$

with

$$\boldsymbol{s}_i = \boldsymbol{Cov}\frac{d^2f}{dx_i d\boldsymbol{x}}\sigma_{xi}^2 \quad . \tag{24}$$

Here, s_i is the coordinate in the direction of $\boldsymbol{s}_i$. Following the principal sensitivity first-order second-moment method (Krüger et al. 2023), these directions $\boldsymbol{s}_i$ are called second-order principal sensitivity directions. A finite difference approximation of $\frac{d^3f}{dx_i d\boldsymbol{y} ds_i}$ is found using (A.10), leading to

$$\sum_{i=1}^{M}\sum_{j=1}^{M} \frac{d^3f}{dx_i dx_j d\boldsymbol{y}} \frac{d^2f}{dx_i dx_j} \cdot \sigma_{xi}^2\sigma_{xj}^2 = \sum_{i}^{M} \frac{\frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i + \epsilon_i \boldsymbol{s}_i) - \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i + \epsilon_i \boldsymbol{s}_i) - \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i) + \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i)}{2\Delta x_i \epsilon_i} \tag{25}$$

with the i-th step-size ϵ_i. This approximation has an error of order $\mathcal{O}(\epsilon_i) + \mathcal{O}(\Delta x_i^2)$. If an error of $\mathcal{O}(\epsilon_i^2) + \mathcal{O}(\Delta x_i^2)$ is required, (A.11) has to be applied, resulting in higher computational cost. Since $\frac{df}{d\boldsymbol{y}}$ has already been evaluated at $(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i)$ and $(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i)$ in (20), only two additional function evaluations have to be done for every value of i. Overall, this framework requires $4M + 1$ function (and gradient) evaluations with the number of random variables M. In consequence, the overall CPU time is $4M + 1$ times the CPU time of the deterministic optimization problem. Based on these considerations, the principal sensitivity second-order fourth-moment method requires comparable (but still higher) computation time to the approach given in Jansen et al. (2015) (which only needs to solve systems of equations instead of the full problem). Assuming that the proposed non-intrusive approach is applied to extremely optimized commercial code instead of implementing the approach from Jansen et al. (2015) by hand, the computation time might be even lower. The overall framework is summarized in Algorithm 1.

Algorithm 1 Principal sensitivity second-order fourth-moment method

1. *Evaluate* f, $\frac{df}{d\boldsymbol{y}}$ *and* $\frac{df}{d\boldsymbol{x}}$ *at mean vector of random variables* $\mu_{\boldsymbol{x}}$
   → *1 function and gradient evaluation*
2. *(Parallel) Evaluate* f, $\frac{df}{d\boldsymbol{y}}$ *and* $\frac{df}{d\boldsymbol{x}}$ *at* $\boldsymbol{x} = \mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i$ *and* $\boldsymbol{x} = \mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i$ *for all* i
   → *2M function and gradient evaluations*
3. *Compute* μ_f, $\frac{d\mu_f}{d\boldsymbol{y}}$ *and* $\sigma^2{}_f$ *using* (19), (20), (9) *and* (21)
   → *no function evaluation*
4. *Compute second-order principal sensitivity directions* $\boldsymbol{s}_i$ *using* (24) *for all* i
   → *no function evaluation*
5. *(Parallel) Evaluate* $\frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} + \boldsymbol{\Delta x}_i + \epsilon_i \boldsymbol{s}_i)$ *and* $\frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}} - \boldsymbol{\Delta x}_i + \epsilon_i \boldsymbol{s}_i)$ *for all* i
   → *2M function and gradient evaluations*
6. *compute* $\frac{d\sigma^2{}_f}{d\boldsymbol{y}}$ *using* (11) *and* (25).
   → *no function evaluation*

The full framework can be strongly parallelized. However, the evaluations of step 5 have to be done subsequent to steps 1 and 2, because the principal sensitivity directions $\boldsymbol{s}_i$ depend on the hessian matrix. If a sufficient number of CPU-cores and memory is available, the computation time is equal to two times the computation time of a deterministic optimization. However, for a high number of random variables, many resources are required.

4.1 Remarks on the step sizes

Choosing a good step size is always a critical problem in the context of finite differences. Unfortunately, there are no general rules that propose a good step size based on model dimensions. Several publications aim at finding a good step size from estimating the truncation error and the round-off error (Wang and Gau 1999; Iott et al. 1985). However, it is usually expected, that the round-off error is caused by rounding of the objective value. This is not always the case for directional derivatives because the step is distributed over many entries leading to significant rounding errors of the step [see discussion in Krüger et al. (2023)].

The first question is which error to track. Mean and variance only depend on analytic terms and on the hessian matrix. Hence, the error of the hessian matrix should be minimized. The gradient of the mean has a consistent error to the mean itself and is therefore not analyzed. In difference to that, the gradient of the variance depends on the numerical approximations of $\frac{d^2f}{dx_i d\boldsymbol{y}}$, $\frac{d^3f}{dx_i d\boldsymbol{y} ds_i}$ as well as on the second-order principal sensitivity direction $\boldsymbol{s}_i$, which depend on the hessian matrix. The numerical studies conducted for this paper have shown that the error of $\frac{d^3f}{dx_i d\boldsymbol{y} ds_i}$ is always the most dominating error. Hence, a step size control should focus on this error. In the following, the expression $\frac{d^3f}{dx_i d\boldsymbol{y} ds_i}$ is called the SO-variance-gradient since it originates form the second-order term in (9).

The SO-variance-gradient is a vector which has as many entries as design variables. From a practical point of view it is not helpful to track all entries of the error vector separately due to the large number of entries. Instead, the error vector is aggregated. In the following, a relative norm error defined as

$$e_{rel} = \frac{\| e\left(\sum_{i=1}^{M} \frac{d^3f}{dx_i d\boldsymbol{y} ds_i} \right) \|}{\| \sum_{i=1}^{M} \frac{d^3f}{dx_i d\boldsymbol{y} ds_i} \|} \tag{26}$$

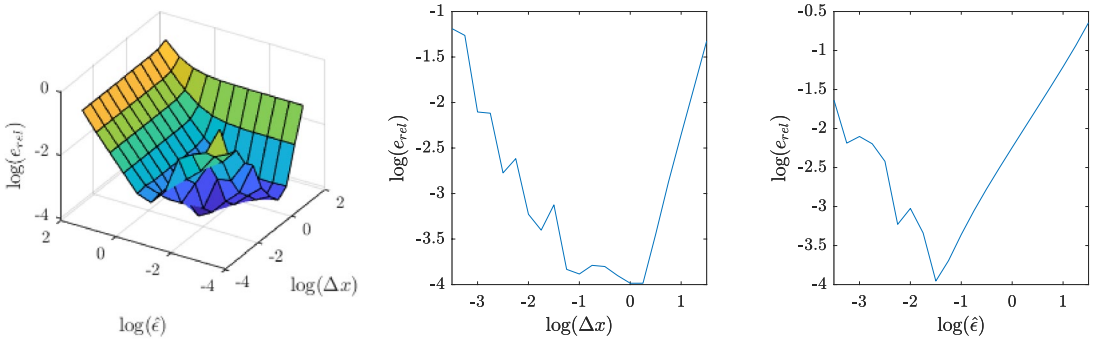
is used, where $e(\boldsymbol{X})$ denotes the error vector of the vector $\boldsymbol{X}$.

Another issue comes with the step $\epsilon_i \boldsymbol{s}_i$ in (25). Since the vector $\boldsymbol{s}_i$ is computed from the hessian matrix of the current design, the length is different for every value of i. Additionally, it changes during the optimization. To circumvent the problem, a normalized step size $\hat{\epsilon}_i$ is used such that

$$\epsilon_i = \frac{\hat{\epsilon}_i}{\|\boldsymbol{s}_i\|} \quad . \tag{27}$$

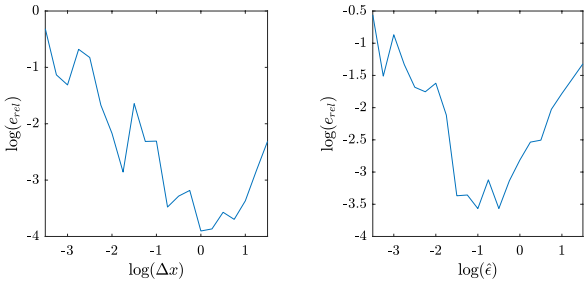
Using this normalization, $\hat{\epsilon}$ has no unit. In theory, it is possible to use different step sizes Δx_i and $\hat{\epsilon}_i$ for every i. However, from a practical point of view, it is hard to optimize many step sizes at the same time. Instead, all step sizes of the same type are chosen to be equal such that $\Delta x_1 = \Delta x_2 = ... = \Delta x_M = \Delta x \neq \hat{\epsilon_1} = \hat{\epsilon_2} = ... = \hat{\epsilon_M} = \hat{\epsilon}$. The distinction between Δx and $\hat{\epsilon}$ is chosen because (25) has a different convergence rate in Δx and $\hat{\epsilon}$.

Figure 1 shows the error of the SO-variance gradient for the initial design of the 2D shallow arc example in Sect. 5.2 with a resolution of 400×30 elements. There, the



(a) 3D plot of error at variable step sizes Δx and $\hat{\epsilon}$. (b) Error at variable step Δx with fixed $\hat{\epsilon} = 0.013$. (c) Error at variable step $\hat{\epsilon}$ with fixed $\Delta x = 0.03$.

Fig. 1 Error of the SO-variance gradient at the later used 2D shallow arch with 12.000 finite elements



(a) Error at variable step Δx with fixed $\hat{\epsilon} = 0.063$. (b) Error at variable step $\hat{\epsilon}$ with fixed $\Delta x = 0.23$.

Fig. 2 Error of the SO-variance gradient at the later used 2D shallow arch with 192.000 finite elements

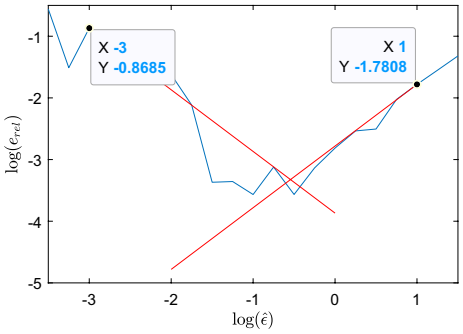


Fig. 3 Step size procedure: optimal step size is found at the intersection of two error lines

error is shown for different step sizes allowing the following observations. Using step sizes $\Delta x = 0.03$ and $\hat{\epsilon} = 0.01$, a relative error of around 0.01% is achieved. At smaller step-sizes, rounding errors occur, while higher step sizes lead to increasing truncation errors. The rounding errors have a slope of 1, while the truncation error has a slope of 2 for Δx and a slope of 1 for $\hat{\epsilon}$. This matches the expectation since Eq. (25) has an error of order $\mathcal{O}(\Delta x^2) + \mathcal{O}(\epsilon)$. The plot for the step size Δx has a plateau in the range of -1.25 to 0.25. This means that the error due to $\hat{\epsilon}$ is dominating there. Similar observations are done for the same model with a much higher resolution of 1600×120 elements. The error of the SO-variance gradient is shown in Fig. 2. The shape of the error is similar to the low-resolution example. However, the best results are found for a much larger step size of $\Delta x = 0.23$ and $\hat{\epsilon} = 0.063$.

4.2 Automatic step size adaption

Without special system knowledge, it is very hard to find good step sizes for the principal sensitivity second-order fourth-moment method. Plotting the error for different step sizes is no good option in practice because this requires a huge amount function evaluations. As an alternative, an automatic step-size adaption scheme is presented. The major goal is to find a good value for Δx and $\hat{\epsilon}$ such that the error of the SO-variance gradient is minimal. An error is caused by truncation in (25), but also by a wrong direction $\boldsymbol{s}_i$ caused by errors in the hessian matrix and round off effects. Especially high errors in $\boldsymbol{s}_i$ lead to unpredictable behavior. Hence, the error of the hessian matrix should be minimal. As a "side effect", a small error in the hessian matrix also leads to accurate approximations of mean and variance. This is achieved by a good step size Δx. The truncation error in (25) is of order $\mathcal{O}(\Delta x)^2\mathcal{O}(\hat{\epsilon})$. Hence, a good step size $\hat{\epsilon}$ leads to a low truncation error.

From the error remainder in (25) it is known that the slope of the logarithmic error of the SO gradient is 1 ($\hat{\epsilon}$) or 2 (Δx) in the truncation area. From Wang and Gau (1999) and Iott et al. (1985) it is known that the rounding error has a slope of approximately 1. In a similar way it is known that the slope of the logarithmic error of the hessian matrix is 2 in the truncation area. If one knows the errors of two points which are in different areas, the optimal step size is approximately found at the intersection of the error lines. This is shown in Fig. 3. Applying the principal sensitivity second-order fourth moment-method mainly makes sense if the SO-variance gradient is not given analytically. In this case, there is no reference to actually compute the error. A usual tool to compute the error if there is no analytic reference is to compare two similar methods with a different convergence rate. To apply this approach to the step size procedure, $\frac{d^3 f}{dx_i \mathbf{dy} ds_i}$ in (25) is now approximated with the higher-order approximation given in (A.11) and the hessian matrix is approximated using finite differences of order 4.

Using the previous considerations, the step size adaption scheme reads as follows:

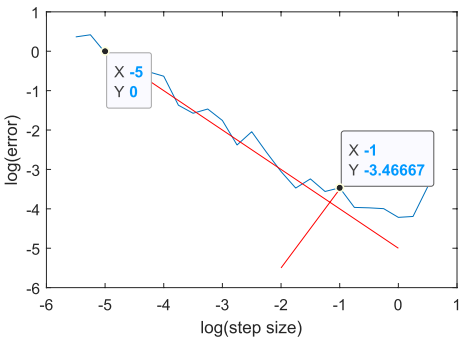


Fig. 4 Possible mistake when using the step size procedure: The initial guess is far away from the optimal step size. In this case, the estimated optimal step near one of the evaluation points

Algorithm 2 automatic step size procedure

1. *choose an initial guess for both step sizes* $\Delta x_0, \hat{\epsilon}_0$
2. *evaluate the error of the hessian matrix: error* e_1 *at* $\Delta x_0 \cdot 10^2$ *and error* e_2 *at* $\Delta x_0 \cdot 10^{-2}$
3. *find optimal step size as intersection of error lines at* $\Delta x_{opt} = \Delta x_0 \cdot 10^{\frac{log(e_2)-log(e_1)+2}{3}}$
4. *evaluate the error of* (25) *at* $\Delta x = \Delta x_{opt}$ *at different values of* $\hat{\epsilon}$*:* e_1 *at* $\hat{\epsilon}_0 \cdot 10^2$ *and* e_2 *at* $\hat{\epsilon}_0 \cdot 10^{-2}$
5. *find optimal step size as intersection of error lines at* $\hat{\epsilon}_{opt} = \hat{\epsilon}_0 \cdot 10^{\frac{log(e_2)-log(e_1)}{2}}$

The step size scheme is built on the assumption that the initial step size Δx_0 is chosen such that $10^{-2}\Delta x_0 < \Delta x_{opt}$ and $10^{-2}\Delta x_0 > \Delta x_{opt}$. If this is not the case, the computed optimal step size is near to $10^{-2}\Delta x_0$ or $10^2\Delta x_0$ (see Fig. 4). In this case, the step size procedure should be repeated with the previously computed steps size as initial guess. The same holds for the step size $\hat{\epsilon}_{opt}$ This step size procedure works robustly on all examples which the authors have tested.

4.3 Matlab implementation

A Matlab implementation including the step size adaption scheme is added as supplementary material. The function "SOFM_PS" computes mean, standard deviation and the gradients using the principal sensitivity second-order fourth-moment method. It strictly follows the Algorithm 1. The function "sofmAdaptStep" is an implementation of the automatic step-size adaption scheme, strictly following Algorithm 2. Both methods are called similar to the Matlab optimization functions using the objective function as an input. Hence, they can be simply applied to many objectives. The main script "SOFM_example.m" demonstrates how to call the functions "SOFM_PS" and "sofmAdaptStep". It includes a short optimization and an error study for a simple objective function.

5 Numeric examples

The presented approach is applied to one shape optimization example and two density-based topology optimization examples using SIMP (Solid Isotropic Material with Penalization) like described in Sigmund (2001). In all examples, the structure is optimized for minimal compliance using geometric nonlinear finite elements under geometric imperfections and volume constraints. In order to show that the method can also be applied to larger models, the last example consists of a 3D shallow arch which is parametrized similar to the first topology optimization example.

All computations are done on a Matlab inhouse tool which mainly uses only one CPU core in parallel. Only the computation of the element stiffness matrices is distibruted over several CPU cores. The geometric nonlinear finite element problems are solved using a load driven path follow solver as described in Wriggers (2008). The material behaviour is modelled with the Saint Venant-Kirchhoff constitutive law. In topology optimization, void elements cause convergence issues for the nonlinear solver. To avoid this problem, a relaxed convergence criterion as well as a continuation scheme for the penalization is used (Buhl et al. 2000) for the topology optimization examples. All occurring systems of equations are solved using a multigrid preconditioned iterative solver as described in Vassilevski (2008) and Amir et al. (2014) with a relative tolerance of 10^{-10}. Since the tangential stiffness matrix becomes indefinite in the postbuckling regime, the GMRES (Saad and Schultz (1986)) solver is used instead of the conjugate gradient method.

The optimization is done using the globally convergent method of moving asymptotes (GCMMA) as optimization method (Svanberg (2002)). The optimization is stopped if the relative change in the objective is less than 10^{-4}. All optimizations start from a homogeneous gray design.

Geometric imperfections are modelled using uncertain nodal coordinates. For that, a random imperfection is added to the vector of nodal coordinates. Correlation is defined using

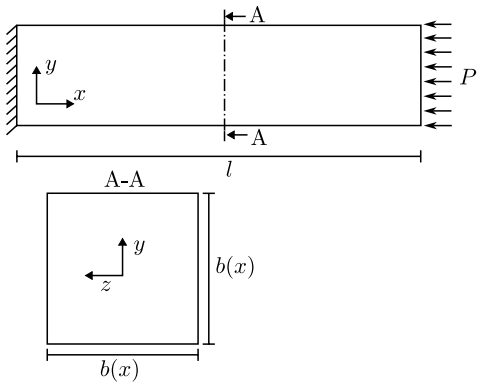


Fig. 5 Model and parameters of the compression bar

length l	$160mm$
initial width $b(x)$	$8mm$
Young's modulus	$2GPa$
Poisson ratio	0.3
filter radius	$80mm$
parameter bounds	$\pm 16mm$
initial move limit	0.1
compression load P	$50N$
number elements	$320 \times 16 \times 16$
correlation length l_c	$80mm$
node variance	$(0.3mm)^2$
objective κ	3

a conditioned random field [see e.g. Ditlevsen (1991)] with a square exponential correlation function. It is assumed that imperfections in different directions are uncorrelated. A spectral decomposition of the random field is done using the EOLE method as described in Sudret and Der Kiureghian (2000).

The overall optimization problem for all examples reads as

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \bar{f}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{u}) \\ \text{s.t.} \quad & \frac{V_0}{V(\boldsymbol{x},\boldsymbol{y})} - 1 \leq 0 \\ & \boldsymbol{y}_{min} \leq \boldsymbol{y} \leq \boldsymbol{y}_{max} \\ & \boldsymbol{0} = \boldsymbol{R}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{u}) \end{aligned} \quad , \tag{28}$$

where the objective function $\bar{f}$ is either the (end-)compliance or the robust objective function (1) of the compliance. Since the problem uses the nonlinear finite element method, the residual $\boldsymbol{R}$ should be zero. In addition, the volume $V(\boldsymbol{x},\boldsymbol{y})$ is restricted to be smaller than an initial volume V_0.

5.1 Compliance shape optimization of an 3D compression bar

The first example consists of a shape optimization of a square-profiled bar under a distributed compression load. The model and its parameters are given in Fig. 5 with the length $l = 160\,\text{mm}$, the position dependent width $b(x) = 8\,\text{mm}$ and the compression load $P = 50\,\text{N}$. The bar is discretized by $320 \times 16 \times 16$ finite elements with the Young's modulus $2GPa$ and the Poissons ratio $\nu = 0.3$. This example corresponds to the first Euler buckling shape (Euler 1744) where the critical load is given by $F_{crit} = \frac{\pi^2 EI}{4l^2}$ with the moment of inertia for a square profile $I = \frac{b^4}{12}$. Inserting the model data of the initial design leads to a critical load of $F_{crit} \approx 65\,\text{N}$.

The shape optimization problem is parametrized by the position-dependent width $b(x)$ at every section of nodes. This is done by moving all surface nodes of the same section simultaneously. As proposed in Le et al. (2011), the inner nodes are updated using Laplacian smoothing to preserve a good mesh quality. In shape optimization, numerical instabilities lead to mesh distortions Le et al. (2011). To avoid this problem, the design variables are filtered similar to the density filter in topology optimization (Sigmund 2007).

Fig. 6 Final designs of the optimization of the compression bar. Top: deterministic optimized design, bottom: robust optimized design

Geometric imperfections are modelled using a square exponential random field with the correlation length $l_c = 80\,\text{mm}$ and nodal variance of $(0.3mm)^2$. In this example, only imperfections in y- and z-direction are considered. To reduce the number of random variables, only the first 20 modes of the corresponding EOLE approximation are considered, leading to a relative error of the variance of less than 0.25%. In the robust optimization, the robust objective function is given by $\bar{f} = \mu_f + \kappa\sigma_f$ with the weighting factor $\kappa = 3$. When using the principal sensitivity second-order fourth moment method, step sizes must be chosen. Using $\Delta x = 2.5 \times 10^{-5}\,\text{m}$ and $\hat{\epsilon} = 2.5 \times 10^{-4}\,\text{m}$, leads to an relative error of the SO gradient of less than 10^{-4}.

The optimizations lead to the designs shown in Fig. 6. The deterministic optimization converges after one iteration because the initial design is already optimal. This is clear since the homogenous compression load is directly transferred to the supports without any curvature effects. However, it is known that this structure is prone to buckling with a critical load slightly above the applied load. Hence, it

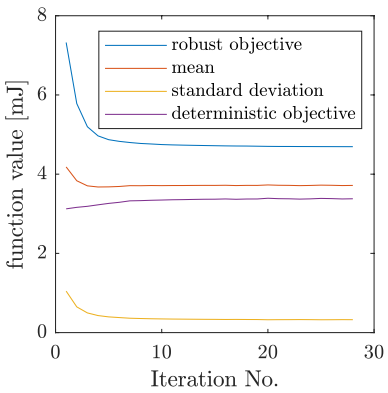


Fig. 7 History of the robust shape optimization of the compression bar

Table 1 Numeric results of the compression bar example

Optimization method →	Deterministic	Robust
f	3.125 mJ	3.379 mJ
μ_f with SOFM	4.192 mJ	3.716 mJ
σ_f with SOFM	1.057 mJ	0.326 mJ
μ_f with MC	4.197 mJ	3.725 mJ
σ_f with MC	1.067 mJ	0.324 mJ
μ_f with FOSM	3.125 mJ	3.379 mJ
σ_f with FOSM	0.019 mJ	0.02 mJ
Time	8.3 s	4700 s
No. iterations	1	28

The stochastic quantities are evaluated using the second-order fourth moment method (SOFM), the first-order second-moment method (FOSM) and the Monte Carlo method (MC) with 10'000 samples

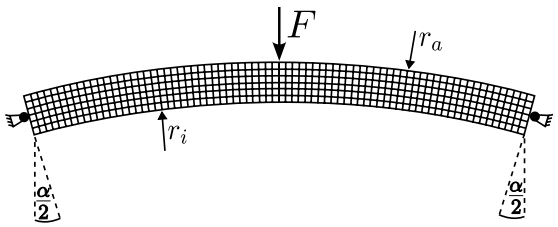
is expected that this structure is not optimal when considering uncertainties.

The robust optimization converges after 28 iterations. The final structure has an increased section area at the support and an decreased section area at the load introduction. This structure has a significant stiffness against bending moments which might occur due to geometric imperfections. The optimization history of the robust optimization is shown in Fig. 7. There, the robust objective as well as mean, standard deviation and deterministic objective function are shown for all iterations. The deterministic result corresponds to the values at iteration 1. The robust objective function follows a perfect asymptotic shape which shows that the optimization runs without any problems. The figure shows that the decrease of mean and standard deviation comes at the price of an increase of the deterministic objective. At iteration 5, the mean starts to increase slightly while the standard deviation still decreases. This is caused by the fact that the standard deviation is weighted by the factor $\kappa = 3$.

In Table 1, the numeric values are shown. There, mean and standard deviation are evaluated using the second-order fourth-moment method (SOFM), the Monte Carlo method using 10,000 samples and the first-order second-moment method (FOSM). Doing a robust design optimization, the mean decreases about 12% and the standard deviation about 70%. This is an huge improvement although the assumed uncertainties lie in a realistic range. In this example, the second-order fourth-moment method excellently approximates the stochastic moments with an error less than 0.25% for the mean and less than 1% for the standard deviation. In difference, the first-order second-moment method significantly underestimates the stochastic quantities for both optimized designs. Especially the standard deviation is underestimated by 90–95% leading to the wrong interpretation that the deterministic optimized design is more robust. This shows the superior accuracy of second-order approaches compared to first-order approaches. Especially it shows that a second-order method is required for the robust optimization to succeed.

Fig. 9 EOLE modes of the 2D shallow arch example. Grey: perfect structure, red: structure with perturbation into corresponding mode direction. (Color figure online)

The computations are done on a workstation with a 64-core AMD Threadripper PRO 5995WX processor. The deterministic optimization was allowed to use 6 cores in parallel while the robust optimization was allowed to use 60 cores. In this example it does not make sense to compare the total computation time since the deterministic optimization only needed 1 iteration. The robust optimization needs $4 \cdot 20 + 1 = 81$ function evaluations that are parallelized to ten workers. Hence it is expected that a robust iteration needs a factor 8 to 9 higher computation time than a deterministic iteration. In reality a factor of 20 is observed. There are several reasons for this observation. First, the power limits of a workstation allow a higher speed per CPU core if less cores are used. Additionally, the CPU speed is higher if the system load is high only for a short time. Second, the FE-system in the first optimization consists of a perfect uniform mesh while the following designs have a non-uniform mesh. Using



α	$30°$
inner radius r_i	$100m$
outer radius r_a	$104m$
thickness t	$1cm$
Young's modulus	$200GPa$
Poisson ratio	0.3
filter radius	$0.25m \cdot \frac{1}{2\sqrt{3}}$
load F	$800kN$
number elements	800×60
volume fraction V_0	30%
penalization p	$1 - 8$
projection η	0
projection β	$1 - 32$
correlation length l_c	$17.45m$
node variance	$(0.33m)^2$
objective κ	1

Fig. 8 Model and parameters of the 2D shallow arch

multigrid preconditioned iterative solvers, the computation time usually increases for non-uniform meshes. In fact, the first robust iteration only needs 107 s which is only a factor 13 higher than the deterministic value.

5.2 Compliance topology optimization of a 2D shallow arch

The second example consists of an two dimensional shallow arch presented in Jansen et al. (2015). The model and its parameters are shown in Fig. 8 with the angle $\alpha = 30°$, the external force $F = 800\,\text{kN}$, the inner radius $r_i = 100\,\text{m}$, the outer radius $r_a = 104\,\text{m}$ and the out of plane thickness $t = 1\,\text{cm}$. The arch is discretized using 800×60 quadrilateral finite elements with the Young's modulus $E = 200\,\text{GPa}$ and the Poisson ratio $\nu = 0.3$. It is loaded by a single load on the middle top node and is simply supported on the left and on the right edge. The simple support is realized by fixing all degrees of freedom of the middle nodes on the corresponding edges.

Imperfections are modelled by a conditioned square exponential random field with a correlation length $l_c = 17.45\,\text{m}$ and a constant variance of $(0.33m)^2$. The random field is conditioned such that the variance is zero for all boundary nodes. To reduce the number of random variables, only the first six modes of a corresponding EOLE approximation are used. The used EOLE modes are illustrated in Fig. 9. There, a perturbation into the direction of the corresponding mode is applied to the deterministically optimized design.

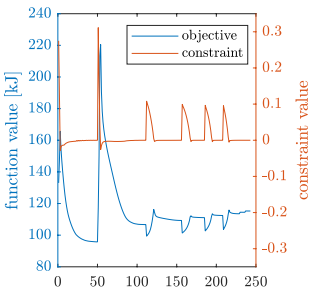
Mesh independency is achieved using the PDE filter given in Lazarov and Sigmund (2011). In order to get black and white designs, a regularized Heaviside projection given by

$$\hat{\rho}_i = \frac{\tanh(\beta\eta) + \tanh(\beta\rho_i - \beta\eta)}{\tanh(\beta\eta) + \tanh(\beta - \beta\eta)} \tag{29}$$

is used. Here, $\hat{\rho}_i$ represents the projected density of element i, which is computed from the filtered density ρ_i. The parameter η is a threshold value and the parameter β defines the steepness of the projection function. In order to enforce a minimum length scale, the threshold value η is set to 0 (Guest et al. 2004; Wang et al. 2011).

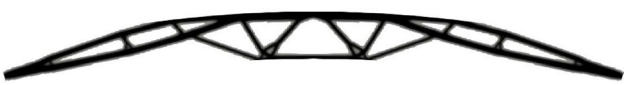
A continuation scheme is applied to the penalization parameter p and to the projection steepness parameter β to reduce the risk of highly deformed elements in the beginning of the optimization. For every set of (p, β), a whole optimization is applied. The final design is afterwards used as the initial design for a following optimization. In detail, the parameters are chosen as $p = i$ and $\beta = \min(32, 2^{i-1})$ for the i-th sequence. Doubling the projection parameter β leads to a violation of the volume constraint. Hence, the optimizer struggles with finding a feasible solution and sometimes produces unstable designs. To circumvent this, the initial move limit is set to $s_{ini} = 0.05$ and the increment for the asymptotes is chosen as $s_{incr} = 1.1$.

(a) Optimized design.

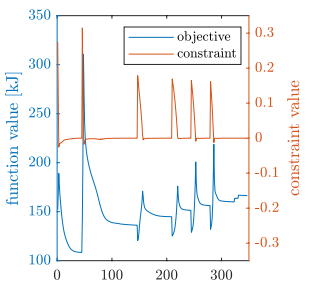


(b) Optimization history.

Fig. 10 Result of the deterministic optimization of the 2D shallow arch



(a) Robust optimized design.



(b) Optimization history.

Fig. 11 Result of the robust optimization of the 2D shallow arch

The deterministic optimization leads to the design shown in Fig. 10a with the optimization history in Fig. 10b. The optimization starts from an infeasible design since a projection with $\eta = 0$ increases the volume. Hence the optimizer is constraint driven in the beginning. Afterwards a perfect asymptotic decrease of the objective is observed. At every peak, the parameters β and p are doubled leading to a constraint violation that has to be fixed.

The final design is similar to the design obtained in Jansen et al. (2015). It can be interpreted as two thin compression bars which are connected to a stiff beam. From engineering knowledge it becomes clear that this structure is sensitive to geometric imperfections, because the bars are prone to buckling.

In the robust optimization, the robust objective function is given by $\bar{f} = \mu_f + \kappa\sigma_f$ with the weighting factor $\kappa = 1$. When using the principal sensitivity second-order fourth-moment method, step sizes have to be chosen. Therefore, the step size adaption scheme with an initial guess of $\Delta x = 10^{-4}, \hat{\epsilon} = 10^{-4}$ is applied twice in the beginning. During the optimization, the step size adaption is applied once after each change of projection and penalization parameters β and p with the last step size as initial guess. In consequence, the error of the SO-variance-gradient is always around 3×10^{-4}.

The optimized design is shown in Fig. 11a with the optimization history in Fig. 11b. The optimization history is similar to the deterministic optimization. Again, the final design is very similar to the design obtained in Jansen et al. (2015). Small differences are observed, which may be caused by a different continuation scheme or a different realization of the boundary conditions, since these aspects are not described in Jansen et al. (2015). Also, different optimization parameters lead to slightly different results. Here, the design can be interpreted as a combination of three beams. The beams are connected in a way that moments are transferred. From engineering knowledge it becomes clear that the robust optimized structure is much less sensitive to shape imperfections since the side beams are able to handle moments. Also asymmetric buckling seems to be unlikely due to the strong connections between the beams.

The previous considerations are supported by the numeric results. The results of both, the deterministic optimization and the robust optimization are summarized in Table 2. There are some aspects to be mentioned. Firstly, the results of the current optimization differ from the results in Jansen et al. (2015) by around 10 %. The difference might be caused by different boundary conditions or load introduction as well as small differences in the design. Large differences are found for the mean and standard deviation of the deterministic optimized design. It is expected that these differences originate from the high sensitivity of the designs with respect to geometric variation. Secondly, the second-order fourth-moment method approximates the stochastic moments very well, when compared to results of a Monte Carlo simulation. At the robust optimized design an error of around 10% for the standard deviation and no error for the mean occurs. The error is comparable to the error obtained by Jansen et. al. In publications from other authors, second-order perturbation approaches have similar errors [see eg. (da Silva and Cardoso 2016; Changizi and Jalalpour 2017; Doltsinis and Kang 2004; Asadpoure et al. 2011)]. In consequence, the robust optimized design shows a significantly improved performance when considering imperfections. Similar to the previous example, the first-order second-moment method totally underestimates mean and standard deviation. Especially in the deterministic case, the standard deviation is underestimated by 99% leading to the wrong interpretation, that the deterministic optimized design is more robust. Again this shows that a second-order method is required for the robust optimization to succeed.

The computations are done on a Laptop with an Intel i7 1185G7 processor. The computation time of the robust optimization is approximately 30 times higher than for the

Table 2 Numeric results of the 2D shallow arch example

Optimization method →	Deterministic	Robust	Deterministic Jansen et al. (2015)	Robust Jansen et al. (2015)
f	108 kJ	121 kJ	97 kJ	121 kJ
μ_f with SOFM	843 kJ	132 kJ	–	126 kJ
σ_f with SOFM	1032 kJ	17 kJ	–	12 kJ
μ_f with MC	705 kJ	132 kJ	337 kJ	127 kJ
σ_f with MC	1820 kJ	19 kJ	1033 kJ	13 kJ
μ_f with FOSM	108 kJ	121 kJ	–	–
σ_f with FOSM	11 kJ	12 kJ	–	–
Computation time	643s	18690s	–	–
No. iterations	243	347	–	–

The stochastic quantities are evaluated using the second-order fourth moment method (SOFM), the first-order second-moment method (FOSM) and the Monte Carlo method (MC) with 10,000 samples

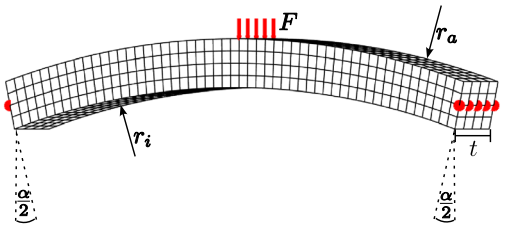
deterministic optimization. This is expected because 6 random parameters lead to $6 \cdot 4 + 1 = 25$ function evaluations per iteration. Parallel computation does not lead to significant improvements due to thermal limits of the laptop.

5.3 Compliance topology optimization of an 3D shallow arch

The last example is a three-dimensional shallow arch consisting of $848 \times 64 \times 64 = 3.5$ M elements. The ratios are similar to the previous example, however, the model is much smaller and has a higher out of plane thickness. The model and its parameters are shown in Fig. 12 with the angle $\alpha = 30°$, the inner radius $r_i = 393$ mm, the outer radius $r_a = 409$ mm and the out of plane thickness $t = 16$ mm. The constant external line force has a value of $F = 25N/16$ mm. The model is supported by a hinge support on the left and right edge. This is modeled by fixing all degrees of freedom on the center line of the corresponding edge. The model is supposed to be 3D printed and hence, the material properties of resin are chosen. It is known that a classic filter forces the material to stick at the boundaries. To circumvent this problem, a PDE filter with consistent boundary conditions like described in Wallin et al. (2020) is used. In this example, a volume preserving projection using $\eta = 0.5$ is applied (Xu et al. 2010).

Again, a continuation scheme is applied to the penalization parameter p and to the projection steepness parameter β. For every set of (p, β), a whole optimization is applied. The parameters are chosen as $p = 0.5 \cdot i + 0.5$ and $\beta = 2^{p-1}$ for the i-th sequence.

Imperfections are modeled with a conditioned square exponential random field with a correlation length $l_c = 70$ mm and a node variance of $(0.5mm)^2$. The random field is conditioned such that the variance at the boundary nodes is zero. In order to reduce the number of random variables, only 15 EOLE modes are considered leading to an error of the variance of less than 3%. The modes are



α	$30°$
inner radius r_i	$393mm$
outer radius r_a	$409mmm$
thickness t	$16mm$
Young's modulus	$2GPa$
Poisson ratio	0.3
Filter radius	$1mm \cdot \frac{1}{2\sqrt{3}}$
line load F	$25N/16mm$
number elements	$848 \times 64 \times 64$
volume fraction V_0	25%
penalization p	$1 - 3$
projection η	0.5
projection β	$1 - 4$
correlation length l_c	$70mm$
Node variance	$(0.5mm)^2$
objective κ	3

Fig. 12 Model and parameters of the 3D shallow arch

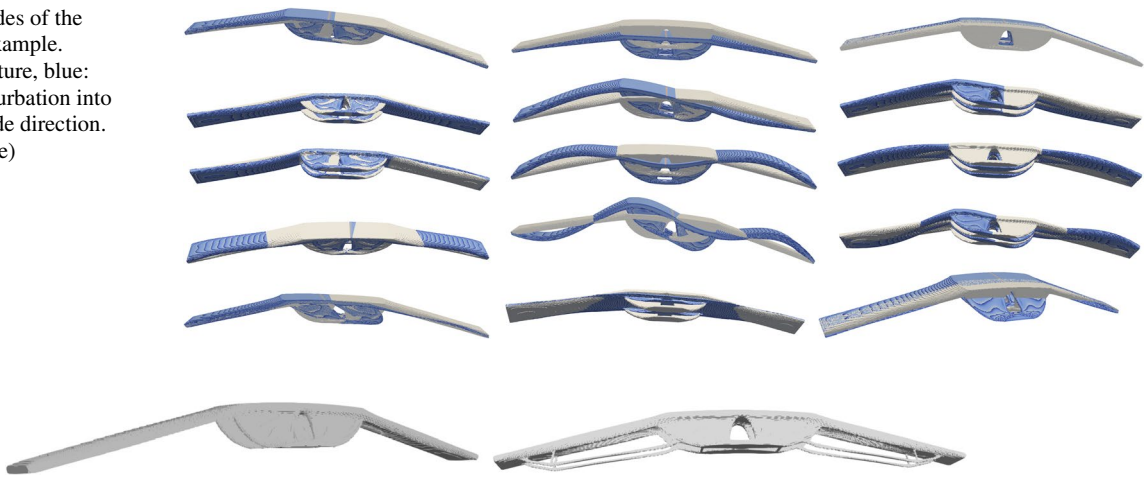


Fig. 13 EOLE modes of the 3D shallow arch example. Grey: perfect structure, blue: structure with perturbation into corresponding mode direction. (Color figure online)

Fig. 14 Final designs of the optimizations of the 3D shallow arc example. Left: Deterministic optimized, right: robust optimized

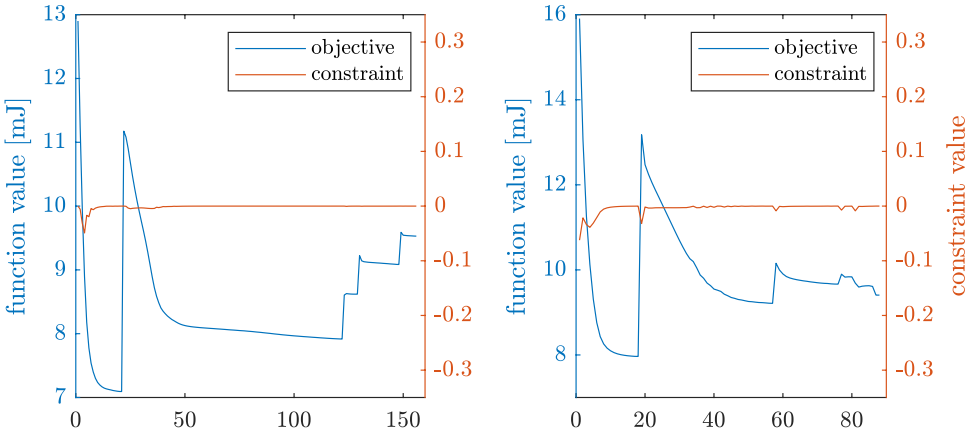


Fig. 15 Optimization history of the 3D shallow arc example. Left: deterministic optimized, right: robust optimized

Table 3 Numeric results of the 3D shallow arch example

Optimization method →	Deterministic	Robust
f	8.16 mJ	8.43 mJ
μ_f with SOFM	8.95 mJ	8.55 mJ
σ_f with SOFM	1.04 mJ	0.34 mJ
μ_f with MC	8.95 mJ	8.60 mJ
σ_f with MC	1.04 mJ	0.36 mJ
μ_f with FOSM	8.16 mJ	8.43 mJ
σ_f with FOSM	0.32 mJ	0.32 mJ
time	19 h	71 h
No. iterations	151	88

The stochastic quantities are evaluated using the second-order fourth moment method (SOFM), the first-order second-moment method (FOSM) and the Monte Carlo method (MC) with 5000 samples

illustrated in Fig. 13. There, a large perturbation into the direction of the corresponding EOLE mode is applied to the deterministically optimized design. This random field seems to be realistic since equivalent tolerances might be categorized as fine in the norm ISO 2768.

For the robust optimization, the robust objective is given by $\bar{f} = \mu_f + \kappa\sigma_f$ with $\kappa = 3$. Assuming that the output function is normal distributed, the choice of κ means that the objective is worse than the robust objective value for less than 0.13% of the samples. Similar to the previous example, the automatic step size procedure is applied after every change of (p, β) leading to an error of the SO-variance gradient of less than 2×10^{-4}.

The optimized designs are shown Fig. 14 with the corresponding optimization histories shown in Fig. 15. The optimization history plots have a similar structure as for the 2D example. At every peak, the parameters β and p are updated. In this example, a volume preserving projection is used and hence the constraint is not violated. However, the parameter update changes the projection and the material penalization leading to worse function values. Between the updates, a perfect asymptotic decrease of the objective values is observed.

The deterministic optimized design is similar to the optimized design in the 2D example. It consists of two plates which are weak against bending and a stiff beam in the middle. This structure is optimal for the given situation because it transforms the (bending) load to a pure compression load in the bars. It uses the full building space and thin plates to achieve a maximum angle between the plates for the external load leading to minimal compression forces. From engineering knowledge it is clear that the plates are prone to buckling.

The robust optimized design significantly differs from the deterministic design. It is also based on the same concept of three parts to transfer the external load to pure compression loads. However, instead of thin plates, beams with a U-profile are used to handle the compression loads. In difference to the plates of the previous design, the beams can handle (small) bending loads. Additionally, the bending-stiff middle part is connected to the beams in a way that (small) bending loads can be transferred to the beams. Thin bars between the stiffener in the middle and the supports further increase the stability. In consequence, the imperfections do not result in buckling of the structure.

The previous interpretations are supported by the numeric values given in Table 3. In the nominal case, the deterministic optimized design has superior performance. However, in presence of imperfections a much worse mean and a high standard deviation are observed. In difference to that, the robust optimized design has a very small standard deviation. The approximation of the stochastic moments using the second-order fourth-moment method is very accurate for the both optimized designs with an error of less than 1% for the mean and less than 6% for the standard deviation. The errors are significantly lower than in the 2D example because a smaller relative standard deviation is used. Again, the first-order second-moment method is much less accurate than the second-order fourth-moment method with the consequence

that the deterministic optimized design would be (wrongly) interpreted as more robust.

Both optimizations are done on a workstation with 2 AMD Epyc 9354 processors. The robust optimization is done on 60 cores, the deterministic optimization is done on 6 cores. At this configuration, the robust optimization needs approximately four times the time of the deterministic optimization. For the robust optimization $15 \cdot 4 + 1 = 61$ function evaluations are required per iteration. The code evaluates 10 functions in parallel leading to the cost of approximately $60/10 = 6$ deterministic optimizations. Considering the higher iteration number of the deterministic optimization, this agrees to the measured time.

6 Conclusions

In the current publication, a new formulation of the second-order fourth-moment method is presented. The number of function evaluations of the objective scales linearly with the number of random parameters and is therefore as effective as the best existing formulations. At the same time it is non-intrusive and hence can be applied to commercial tools without significant effort. The proposed approach uses finite difference approximations, which lead to numerical errors. It is shown that the error depends on two different step sizes Δx and $\hat{\epsilon}$ with an order of $\mathcal{O}(\Delta x^2) + \mathcal{O}(\hat{\epsilon})$. Without further knowledge it is hard to find good step sizes. To circumvent this problem, an effective automatic step size procedure is provided which leads to nearly optimal steps in the examples considered. The approach is applied to different examples showing that it leads to the same results as given in literature, which shows that the numeric errors do not disturb the optimization significantly. Even in a 3D example with 10 million degrees of freedom, the approximation provides a good accuracy at moderate computational cost leading to good results.

Future research might address the combination of the proposed method with the principal sensitivity first-order second-moment method. That way, the most important modes might be tackled using the second-order fourth-moment while the (high frequent) rest of the random field is considered using the less accurate first-order second-moment method. Hence, many random parameters can be considered without the need of a reduction of the random field at low computational cost.

Appendix A: Derivation of finite difference approximations for second-order derivatives

For the derivation of finite difference approximations, the objective function is expanded using different Taylor series given by

$$T_1 = f(\boldsymbol{x} + \boldsymbol{\Delta x}_1) = f(\boldsymbol{x}) + \frac{df}{dx_1}\Delta x_1 + \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 + \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 + \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 + \mathcal{O}(\Delta x_1^5) \tag{A.1}$$

$$T_2 = f(\boldsymbol{x} - \boldsymbol{\Delta x}_1) = f(\boldsymbol{x}) - \frac{df}{dx_1}\Delta x_1 + \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 - \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 + \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 + \mathcal{O}(\Delta x_1^5) \tag{A.2}$$

$$T_3 = f(\boldsymbol{x} + \boldsymbol{\Delta x}_2) = f(\boldsymbol{x}) + \frac{df}{dx_2}\Delta x_2 + \frac{1}{2}\frac{d^2f}{dx_2^2}\Delta x_2^2 + \frac{1}{6}\frac{d^3f}{dx_2^3}\Delta x_2^3 + \frac{1}{24}\frac{d^4f}{dx_2^4}\Delta x_2^4 + \mathcal{O}(\Delta x_2^5) \tag{A.3}$$

$$\begin{aligned} T_4 = f(\boldsymbol{x} + \boldsymbol{\Delta x}_1 + \boldsymbol{\Delta x}_2) = f(\boldsymbol{x}) &+ \frac{df}{dx_1}\Delta x_1 + \frac{df}{dx_2}\Delta x_2 \\ &+ \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 + \frac{d^2f}{dx_1 dx_2}\Delta x_1\Delta x_2 \\ &+ \frac{1}{2}\frac{d^2f}{dx_2^2}\Delta x_2^2 + \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 + \frac{1}{2}\frac{d^3f}{dx_1^2 dx_2}\Delta x_1^2\Delta x_2 \\ &+ \frac{1}{2}\frac{d^3f}{dx_1 dx_2^2}\Delta x_1\Delta x_2^2 + \frac{1}{6}\frac{d^3f}{dx_2^3}\Delta x_2^3 \\ &+ \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 + \frac{1}{6}\frac{d^4f}{dx_1^3 dx_2}\Delta x_1^3\Delta x_2 + \frac{1}{4}\frac{d^4f}{dx_1^2 dx_2^2}\Delta x_1^2\Delta x_2^2 \\ &+ \frac{1}{6}\frac{d^4f}{dx_1 dx_2^3}\Delta x_1\Delta x_2^3 + \frac{1}{24}\frac{d^4f}{dx_2^4}\Delta x_2^4 \\ &+ \mathcal{O}(\Delta x_1^5) + \mathcal{O}(\Delta x_2^5) + \mathcal{O}(\Delta x_1^3\Delta x_2^2) + \mathcal{O}(\Delta x_1^2\Delta x_2^3) \\ &+ \mathcal{O}(\Delta x_1^4\Delta x_2) + \mathcal{O}(\Delta x_1\Delta x_2^4) \end{aligned} \tag{A.4}$$

$$
\begin{aligned}
T_5 = f(\boldsymbol{x} - \boldsymbol{\Delta x}_1 + \boldsymbol{\Delta x}_2) &= f(\boldsymbol{x}) - \frac{df}{dx_1}\Delta x_1 + \frac{df}{dx_2}\Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 - \frac{d^2f}{dx_1 dx_2}\Delta x_1 \Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_2^2}\Delta x_2^2 - \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 + \frac{1}{2}\frac{d^3f}{dx_1^2 dx_2}\Delta x_1^2 \Delta x_2 \\
&- \frac{1}{2}\frac{d^3f}{dx_1 dx_2^2}\Delta x_1 \Delta x_2^2 + \frac{1}{6}\frac{d^3f}{dx_2^3}\Delta x_2^3 \\
&+ \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 - \frac{1}{6}\frac{d^4f}{dx_1^3 dx_2}\Delta x_1^3 \Delta x_2 + \frac{1}{4}\frac{d^4f}{dx_1^2 dx_2^2}\Delta x_1^2 \Delta x_2^2 \\
&- \frac{1}{6}\frac{d^4f}{dx_1 dx_2^3}\Delta x_1 \Delta x_2^3 + \frac{1}{24}\frac{d^4f}{dx_2^4}\Delta x_2^4 \\
&+ \mathcal{O}(\Delta x_1^5) + \mathcal{O}(\Delta x_2^5) + \mathcal{O}(\Delta x_1^3 \Delta x_2^2) + \mathcal{O}(\Delta x_1^2 \Delta x_2^3) \\
&+ \mathcal{O}(\Delta x_1^4 \Delta x_2) + \mathcal{O}(\Delta x_1 \Delta x_2^4)
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
T_6 = f(\boldsymbol{x} + \boldsymbol{\Delta x}_1 - \boldsymbol{\Delta x}_2) &= f(\boldsymbol{x}) + \frac{df}{dx_1}\Delta x_1 - \frac{df}{dx_2}\Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 - \frac{d^2f}{dx_1 dx_2}\Delta x_1 \Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_2^2}\Delta x_2^2 + \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 - \frac{1}{2}\frac{d^3f}{dx_1^2 dx_2}\Delta x_1^2 \Delta x_2 \\
&+ \frac{1}{2}\frac{d^3f}{dx_1 dx_2^2}\Delta x_1 \Delta x_2^2 - \frac{1}{6}\frac{d^3f}{dx_2^3}\Delta x_2^3 \\
&+ \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 - \frac{1}{6}\frac{d^4f}{dx_1^3 dx_2}\Delta x_1^3 \Delta x_2 + \frac{1}{4}\frac{d^4f}{dx_1^2 dx_2^2}\Delta x_1^2 \Delta x_2^2 \\
&- \frac{1}{6}\frac{d^4f}{dx_1 dx_2^3}\Delta x_1 \Delta x_2^3 + \frac{1}{24}\frac{d^4f}{dx_2^4}\Delta x_2^4 \\
&+ \mathcal{O}(\Delta x_1^5) + \mathcal{O}(\Delta x_2^5) + \mathcal{O}(\Delta x_1^3 \Delta x_2^2) + \mathcal{O}(\Delta x_1^2 \Delta x_2^3) \\
&+ \mathcal{O}(\Delta x_1^4 \Delta x_2) + \mathcal{O}(\Delta x_1 \Delta x_2^4)
\end{aligned}
\tag{A.6}
$$

$$
\begin{aligned}
T_7 = f(\boldsymbol{x} - \boldsymbol{\Delta x}_1 - \boldsymbol{\Delta x}_2) &= f(\boldsymbol{x}) - \frac{df}{dx_1}\Delta x_1 - \frac{df}{dx_2}\Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_1^2}\Delta x_1^2 + \frac{d^2f}{dx_1 dx_2}\Delta x_1 \Delta x_2 \\
&+ \frac{1}{2}\frac{d^2f}{dx_2^2}\Delta x_2^2 - \frac{1}{6}\frac{d^3f}{dx_1^3}\Delta x_1^3 - \frac{1}{2}\frac{d^3f}{dx_1^2 dx_2}\Delta x_1^2 \Delta x_2 \\
&- \frac{1}{2}\frac{d^3f}{dx_1 dx_2^2}\Delta x_1 \Delta x_2^2 - \frac{1}{6}\frac{d^3f}{dx_2^3}\Delta x_2^3 \\
&+ \frac{1}{24}\frac{d^4f}{dx_1^4}\Delta x_1^4 + \frac{1}{6}\frac{d^4f}{dx_1^3 dx_2}\Delta x_1^3 \Delta x_2 + \frac{1}{4}\frac{d^4f}{dx_1^2 dx_2^2}\Delta x_1^2 \Delta x_2^2 \\
&+ \frac{1}{6}\frac{d^4f}{dx_1 dx_2^3}\Delta x_1 \Delta x_2^3 + \frac{1}{24}\frac{d^4f}{dx_2^4}\Delta x_2^4 \\
&+ \mathcal{O}(\Delta x_1^5) + \mathcal{O}(\Delta x_2^5) + \mathcal{O}(\Delta x_1^3 \Delta x_2^2) + \mathcal{O}(\Delta x_1^2 \Delta x_2^3) \\
&+ \mathcal{O}(\Delta x_1^4 \Delta x_2) + \mathcal{O}(\Delta x_1 \Delta x_2^4)
\end{aligned}
\tag{A.7}
$$

For the homogeneous second-order derivative, central differences given by

$$
\frac{d^2f}{dx_1^2} = \frac{T_1 + T_2 - 2f(\boldsymbol{x})}{\Delta x_1^2} + \mathcal{O}(\Delta x_1^2) \tag{A.8}
$$

are used. This method has convergence order two with only two additional function evaluations. The computation of the mixed derivatives is more advanced.

A method with convergence order one is given by

$$
\begin{aligned}
\frac{d^2f}{dx_1 dx_2} &= \frac{T_4 - T_3 - T_1 + f(\boldsymbol{x})}{\Delta x_1 \Delta x_2} - \frac{1}{2}\frac{d^3f}{dx_1^2 dx_2}\Delta x_1 \\
&\quad - \frac{1}{2}\frac{d^3f}{dx_1 dx_2^2}\Delta x_2 \\
&\quad + \mathcal{O}\left(\frac{\Delta x_1^3}{\Delta x_2}\right) + \mathcal{O}(\Delta x_1^2) + \mathcal{O}(\Delta x_2^2) + \mathcal{O}\left(\frac{\Delta x_2^3}{\Delta x_1}\right) \\
&= \frac{T_4 - T_3 - T_1 + f(\boldsymbol{x})}{\Delta x_1 \Delta x_2} + \mathcal{O}(\Delta x_1) + \mathcal{O}(\Delta x_2)
\end{aligned}
\tag{A.9}
$$

With only one additional function evaluation, a method is derived which has a convergence order of two regarding the step size Δx_1 and a convergence order of one regarding the step size Δx_2. This is done using

$$\begin{aligned}\frac{d^2f}{dx_1dx_2} &= \frac{T_4+T_2-T_5-T_1}{2\Delta x_1\Delta x_2} - \frac{1}{2}\frac{d^3f}{dx_1dx_2^2}\Delta x_2 \\ &\quad - \frac{1}{6}\frac{d^4f}{dx_1^3dx_2}\Delta x_1^2 - \frac{1}{6}\frac{d^4f}{dx_1dx_2^3}\Delta x_2^2 \\ &\quad + \mathcal{O}(\frac{\Delta x_1^4}{\Delta x_2}) + \mathcal{O}(\Delta x_1^3) + \mathcal{O}(\Delta x_1^2\Delta x_2) \\ &\quad + \mathcal{O}(\Delta x_1\Delta x_2^2) \\ &\quad + \mathcal{O}(\Delta x_2^3) + \mathcal{O}(\frac{\Delta x_2^4}{\Delta x_1}) \\ &= \frac{T_4+T_2-T_5-T_1}{2\Delta x_1\Delta x_2} + \mathcal{O}(\Delta x_2) + \mathcal{O}(\Delta x_1^2)\end{aligned} \tag{A.10}$$

A method with second-order convergence for both step sizes is given by

$$\begin{aligned}\frac{d^2f}{dx_1dx_2} &= \frac{T_4+T_7-T_5-T_6}{4\Delta x_1\Delta x_2} \\ &\quad - \frac{1}{6}\frac{d^4f}{dx_1^3dx_2}\Delta x_1^2 - \frac{1}{6}\frac{d^4f}{dx_1dx_2^3}\Delta x_2^2 \\ &\quad + \mathcal{O}(\frac{\Delta x_1^4}{\Delta x_2}) + \mathcal{O}(\Delta x_1^3) + \mathcal{O}(\Delta x_1^2\Delta x_2) \\ &\quad + \mathcal{O}(\Delta x_1\Delta x_2^2) \\ &\quad + \mathcal{O}(\Delta x_2^3) + \mathcal{O}(\frac{\Delta x_2^4}{\Delta x_1}) \\ &= \frac{T_4+T_7-T_5-T_6}{4\Delta x_1\Delta x_2} + \mathcal{O}(\Delta x_1^2) + \mathcal{O}(\Delta x_2^2)\end{aligned} \tag{A.11}$$

Appendix B: Approach by Jansen et al.

In Jansen et al. (2015), the robust objective f_R is computed by inserting the derivatives given by

$$\frac{df}{dx_i} = \frac{\partial f}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{dx_i} \tag{B12}$$

$$\begin{aligned}\frac{d^2f}{dx_idx_j} &= \frac{d\boldsymbol{u}}{dx_i}^T\frac{\partial^2 f}{\partial \boldsymbol{u}^2}\frac{d\boldsymbol{u}}{dx_j} \\ &\quad - \boldsymbol{v}^T\left(\frac{\partial \boldsymbol{K}_T}{\partial x_i}\frac{d\boldsymbol{u}}{dx_j} + \frac{\partial \boldsymbol{K}_T}{\partial x_j}\frac{d\boldsymbol{u}}{dx_i} + \frac{\partial^2 \boldsymbol{R}}{\partial x_i\partial x_j} + (\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{dx_i})\frac{d\boldsymbol{u}}{dx_j}\right)\end{aligned} \tag{B13}$$

into (3) and (4). Here, $\boldsymbol{R}$ represents the vector of inner forces and $\boldsymbol{K}_T$ the tangential stiffness matrix. To compute the state variables the nonlinear systems of equations

$$\boldsymbol{R}(\boldsymbol{u}) - \boldsymbol{P} = 0 \tag{B14}$$

$$\boldsymbol{K}_T\frac{d\boldsymbol{u}}{dx_i} + \frac{\partial \boldsymbol{R}}{dx_i} = 0 \quad i = 1, ..., M \tag{B15}$$

$$\boldsymbol{K}_T\boldsymbol{v} - \frac{\partial f}{\partial \boldsymbol{u}} = 0 \tag{B16}$$

with the external load vector $\boldsymbol{P}$ are solved. Here, M represents the number of random variables. The state variables $\boldsymbol{u}$, $\frac{d\boldsymbol{u}}{dx_i}$ and $\boldsymbol{v}$ represent the model deformation, the derivatives of the model deformation and an adjoint variable respectively. In order to compute the gradient of the robust objective, the robust objective f_R is augmented using the previous Eqs. (B14)–(B16). This leads to the Lagrangian

$$\begin{aligned}\hat{f}_R &= f_R - \lambda_0^T(\boldsymbol{R}-\boldsymbol{P}) - \sum_{i=1}^{M}\lambda_i^T\left(\boldsymbol{K}_T\frac{d\boldsymbol{u}}{dx_i} + \frac{\partial \boldsymbol{R}}{\partial x_i}\right) \\ &\quad - \boldsymbol{\gamma}^T\left(\boldsymbol{K}_T\boldsymbol{v} - \frac{\partial f}{\partial \boldsymbol{u}}\right)\end{aligned} \tag{B17}$$

with the adjoint variables λ_0, λ_i and $\boldsymbol{\gamma}$. Differentiation with respect to the design variables $\boldsymbol{y}$ leads to

$$\begin{aligned}\frac{df_R}{d\boldsymbol{y}} &= \frac{\partial f}{\partial \boldsymbol{y}} - \lambda_0^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} - \sum_{i=1}^{M}\lambda_i^T\left(\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{y}}\frac{d\boldsymbol{u}}{dx_i} + \frac{\partial^2 \boldsymbol{R}}{\partial x_i\partial \boldsymbol{y}}\right) \\ &\quad - \boldsymbol{\gamma}^T\left(\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{y}}\boldsymbol{v} - \frac{\partial^2 f}{\partial \boldsymbol{u}\partial \boldsymbol{y}}\right)\end{aligned} \tag{B18}$$

with the adjoint systems

$$0 = \boldsymbol{K}_T\boldsymbol{\gamma} - \frac{\partial f_R}{\partial \boldsymbol{v}}^T \tag{B19}$$

$$0 = \boldsymbol{K}_T\lambda_i - \frac{\partial f_R}{\partial \frac{d\boldsymbol{u}}{dx_i}}^T \quad i = 1, ..., M \tag{B20}$$

$$0 = \boldsymbol{K}_T\lambda_0 - \frac{\partial f_R}{\partial \boldsymbol{u}}^T + \sum_{i=1}^{M}\frac{\partial \boldsymbol{K}_T}{\partial x_i}\lambda_i + \left(\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{u}}\boldsymbol{v}\right)^T\boldsymbol{\gamma} \tag{B21}$$

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—508865334.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors state that there is no conflict of interest.

Replication of results The authors state that the paper contains all information necessary to reproduce the results.

References

Amir O, Aage N, Lazarov BS (2014) On multigrid-CG for efficient topology optimization. Struct Multidisc Optim 49(5):815–829. https://doi.org/10.1007/s00158-013-1015-5

Asadpoure A, Tootkaboni M, Guest JK (2011) Robust topology optimization of structures with uncertainties in stiffness—application to truss structures. Comput Struct 89(11):1131–1141. https://doi.org/10.1016/j.compstruc.2010.11.004

Beyer H-G, Sendhoff B (2007) Robust optimization—a comprehensive survey. Comput Methods Appl Mech Eng 196(33):3190–3218. https://doi.org/10.1016/j.cma.2007.03.003

Buhl T, Pedersen CBW, Sigmund O (2000) Stiffness design of geometrically nonlinear structures using topology optimization. Struct Multidisc Optim 19(2):93–104. https://doi.org/10.1007/s001580050089

Changizi N, Jalalpour M (2017) Robust topology optimization of frame structures under geometric or material properties uncertainties. Struct Multidisc Optim 56(4):791–807. https://doi.org/10.1007/s00158-017-1686-4

Chatterjee T, Chakraborty S, Chowdhury R (2019) A critical review of surrogate assisted robust design optimization. Arch Comput Methods Eng 26(1):245–274. https://doi.org/10.1007/s11831-017-9240-5

Ditlevsen O (1991) Random field interpolation between point by point measured properties. In: Spanos PD, Brebbia CA (eds) Computational stochastic mechanics. Springer, Dordrecht, pp 801–812. https://doi.org/10.1007/978-94-011-3692-1_67

Doltsinis I, Kang Z (2004) Robust design of structures using optimization methods. Comput Methods Appl Mech Eng 193(23):2221–2237. https://doi.org/10.1016/j.cma.2003.12.055

Doltsinis I, Kang Z, Cheng G (2005) Robust design of non-linear structures using optimization methods. Comput Methods Appl Mech Eng 194(12):1779–1795. https://doi.org/10.1016/j.cma.2004.02.027

Euler L (1744) Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive solutio problematis isoperimetrici lattissimo sensu accepti. Marcum-Michaelem Bousquet, Lausanne, pp 1–322

Green L, Lin H-Z, Khalessi M (2002) Probabilistic methods for uncertainty propagation applied to aircraft design. In: 20th AIAA applied aerodynamics conference, pp 1–18. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2002-3140. Accessed 12 Aug 2023

Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int J Numer Methods Eng 61(2):238–254. https://doi.org/10.1002/nme.1064

Haldar A, Mahadevan S (1999) Probability, reliability and statistical methods in engineering design, 1st edn. Wiley, New York

Hederberg H, Thore C-J (2023) Worst-case compliance for independently constrained uncertain loads. Comput Struct 289:107178. https://doi.org/10.1016/j.compstruc.2023.107178

Helton JC, Johnson JD, Oberkampf WL (2004) An exploration of alternative approaches to the representation of uncertainty in model predictions. Reliab Eng Syst Saf 85(1):39–71. https://doi.org/10.1016/j.ress.2004.03.025

Henrichsen SR, Lindgaard E, Lund E (2015) Robust buckling optimization of laminated composite structures using discrete material optimization considering worst shape imperfections. Thin-Walled Struct 94:624–635. https://doi.org/10.1016/j.tws.2015.05.004

Iott J, Haftka RT, Adelman HM (1985) Selecting step sizes in sensitivity analysis by finite differences. Technical Report NASA-TM-86382, NASA. https://ntrs.nasa.gov/citations/19850025225. Accessed 6 Mar 2023

Isserlis L (1918) On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. Biometrika 12(1/2):134–139

Jansen M, Lombaert G, Schevenels M (2015) Robust topology optimization of structures with imperfect geometry based on geometric nonlinear analysis. Comput Methods Appl Mech Eng 285:452–467. https://doi.org/10.1016/j.cma.2014.11.028

Kanno Y (2020) On three concepts in robust design optimization: absolute robustness, relative robustness, and less variance. Struct Multidisc Optim 62(2):979–1000. https://doi.org/10.1007/s00158-020-02503-9

Keshavarzzadeh V, Fernandez F, Tortorelli DA (2017) Topology optimization under uncertainty via non-intrusive polynomial chaos expansion. Comput Methods Appl Mech Eng 318:120–147. https://doi.org/10.1016/j.cma.2017.01.019

Kiureghian AD (2022) Structural and system reliability. Cambridge University Press. https://doi.org/10.1017/9781108991889 . https://www.cambridge.org/highereducation/books/structural-and-system-reliability/7B7F299239AD41812A0C3E2E93B3CA57 Accessed 11 Dec 2023

Kranz M, Lüdeker JK, Kriegesmann B (2023) A generalized approach for robust topology optimization using the first-order second-moment method for arbitrary response functions. Struct Multidisc Optim 66(5):98. https://doi.org/10.1007/s00158-023-03540-w

Kriegesmann B (2021) On the applicability of first-order approximations for design optimization under uncertainty. In: Challamel N, Kaplunov J, Takewaki I (eds) Modern trends in structural and solid mechanics 3—non-deterministic mechanics. Wiley, Hoboken, pp 39–60. https://doi.org/10.1002/9781119831839.ch3

Kriegesmann B, Lüdeker JK (2019) Robust compliance topology optimization using the first-order second-moment method. Struct Multidisc Optim 60(1):269–286. https://doi.org/10.1007/s00158-019-02216-8

Kriegesmann B, Rolfes R, Hühne C, Kling A (2011) Fast probabilistic design procedure for axially compressed composite cylinders. Compos Struct 93:3140–3149. https://doi.org/10.1016/j.compstruct.2011.06.017

Krüger JC, Kriegesmann B (2024) Efficient robust topology optimization of eigenfrequencies using the first-order second-moment method. In: Nachbagauer K, Held A (eds) Optimal design and control of multibody systems. IUTAM Bookseries. Springer, Cham, pp 75–84. https://doi.org/10.1007/978-3-031-50000-8_7

Krüger JC, Kranz M, Schmidt T, Seifried R, Kriegesmann B (2023) An efficient and non-intrusive approach for robust design optimization with the first-order second-moment method. Comput Methods Appl Mech Eng 414:116136. https://doi.org/10.1016/j.cma.2023.116136

Lazarov BS, Schevenels M, Sigmund O (2012) Topology optimization with geometric uncertainties by perturbation techniques.

Int J Numer Methods Eng 90(11):1321–1336. https://doi.org/10.1002/nme.3361
Lazarov BS, Sigmund O (2011) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Methods Eng 86(6):765–781. https://doi.org/10.1002/nme.3072
Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 200(9):985–996. https://doi.org/10.1016/j.cma.2010.10.004
Mircea Grigoriu (1998) Simulation of stationary non-Gaussian translation processes. J Eng Mech 124(2):121–126. https://doi.org/10.1061/(ASCE)0733-9399(1998)124:2(121)
Mäck M, Caylak I, Edler P, Freitag S, Hanss M, Mahnken R, Meschke G, Penner E (2019) Optimization with constraints considering polymorphic uncertainties. GAMM-Mitteilungen 42(1):201900005. https://doi.org/10.1002/gamm.201900005
Papaioannou I, Daub M, Drieschner M, Duddeck F, Ehre M, Eichner L, Eigel M, Götz M, Graf W, Grasedyck L, Gruhlke R, Hömberg D, Kaliske M, Moser D, Petryna Y, Straub D (2019) Assessment and design of an engineering structure with polymorphic uncertainty quantification. GAMM-Mitteilungen 42(2):201900009. https://doi.org/10.1002/gamm.201900009
Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7(3):856–869. https://doi.org/10.1137/0907058
Sartorti R, Möcker T, Kriegesmann B, Pedersen CBW (2023) On non-parametric fatigue optimization. Int J Numer Methods Eng 124(5):1168–1192. https://doi.org/10.1002/nme.7158
Schevenels M, Lazarov BS, Sigmund O (2011) Robust topology optimization accounting for spatially varying manufacturing errors. Comput Methods Appl Mech Eng 200(49–52):3613–3627. https://doi.org/10.1016/j.cma.2011.08.006
Schuëller GI, Valdebenito MA (2010) Reliability-based optimization—an overview. Comput Technol Rev 1:121–155
Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidisc Optim 21(2):120–127
Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33(4–5):401–424. https://doi.org/10.1007/s00158-006-0087-x
Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Struct Multidisc Optim 43(5):589–596. https://doi.org/10.1007/s00158-011-0638-7
Silva GA, Cardoso EL (2016) Topology optimization of continuum structures subjected to uncertainties in material properties. Int J Numer Methods Eng 106(3):192–212. https://doi.org/10.1002/nme.5126
Steltner K, Pedersen CBW, Kriegesmann B (2022) Semi-intrusive approach for stiffness and strength topology optimization under uncertainty. Optim Eng. https://doi.org/10.1007/s11081-022-09770-z
Sudret B, Der Kiureghian A (2000) Stochastic finite element methods and reliability: a state-of-the-art report. Technical Report UCB/SEMM-2000/08, Department of Civil and Environmental Engineering, University of California Berkeley, Berkeley, California
Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12(2):555–573. https://doi.org/10.1137/S1052623499362822
Tan J, Faghihi D (2024) A scalable framework for multi-objective PDE-constrained design of building insulation under uncertainty. Comput Methods Appl Mech Eng 419:116628. https://doi.org/10.1016/j.cma.2023.116628
Torres AP, Warner JE, Aguiló MA, Guest JK (2021) Robust topology optimization under loading uncertainties via stochastic reduced order models. Int J Numer Methods Eng 122(20):5718–5743. https://doi.org/10.1002/nme.6770
Valm N, Changizi N, Tootkaboni M, Asadpoure A (2022) Topology optimization of imperfect frame structures with improved manufacturability. Int J Mech Sci 213:106869. https://doi.org/10.1016/j.ijmecsci.2021.106869
Vassilevski PS (2008) Multilevel block factorization preconditioners. Springer, New York. https://doi.org/10.1007/978-0-387-71564-3 . http://link.springer.com/10.1007/978-0-387-71564-3. Accessed 29 Aug 2023
Wallin M, Ivarsson N, Amir O, Tortorelli D (2020) Consistent boundary conditions for PDE filter regularization in topology optimization. Struct Multidisc Optim. https://doi.org/10.1007/s00158-020-02556-w
Wang L, Gau K (1999) Automatic step-size procedure in forward-difference for reliability and design optimization. In: Proceedings of the 25th design automation conference, pp 97–105. American Society of Mechanical Engineers Digital Collection. https://doi.org/10.1115/DETC99/DAC-8603 . https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-abstract/DETC99/19715/97/1097298. Accessed 6 Mar 2023
Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Struct Multidisc Optim 43(6):767–784. https://doi.org/10.1007/s00158-010-0602-y
Wriggers P (2008) Nonlinear finite element method, Springer. https://doi.org/10.1007/978-3-540-71001-1_1 . http://link.springer.com/chapter/10.1007/978-3-540-71001-1_1. Accessed 10 Oct 2016
Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside functions. Struct Multidisc Optim 41(4):495–505. https://doi.org/10.1007/s00158-009-0452-7
Yao W, Chen X, Luo W, Tooren M, Guo J (2011) Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. Progress Aerosp Sci 47(6):450–479. https://doi.org/10.1016/j.paerosci.2011.05.001