Scalable gradient-based approaches for topology and shape optimization under uncertainty

Vom Promotionsausschuss der Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing)

genehmigte Dissertation (kumulativ)

von
Jan Christoph Krüger

aus
Elmshorn

2026

Erster Gutachter: Prof. Dr.-Ing Benedikt Kriegesmann
Zweiter Gutachter Prof. Dr.-Ing Thomas Rung

Tag der mündlichen Prüfung: 17. April 2026
DOI: 10.15480/882.17010
ORCiD: 0000-0002-4080-9379

Abstract

The performance of mechanical components can be significantly improved using structural optimization approaches, such as topology or shape optimization. However, optimized designs are often tailored to idealized conditions, which potentially leads to degraded performance under imperfect scenarios. Using robust design optimization (RDO), a linear combination of stochastic moments is optimized instead of deterministic function values. Therefore, a probabilistic analysis is embedded into the optimization framework leading to less sensitive designs. However, most robust design optimization frameworks come at high computational cost limiting their scalability.

The current dissertation addresses the challenge of finding scalable robust optimization frameworks. The focus is therefore set on Taylor series-based methods, namely the first-order second-moment method (FOSM) and the more accurate second-order fourth-moment method (SOFM). These methods approximate the stochastic moments using a first-order (FOSM) or second-order (SOFM) Taylor approximation of the objective function. In the context of high-dimensional structural optimization problems, the gradients of the stochastic moments are required. While existing formulations require either a large number of system solves or memory-consuming and highly intrusive partial derivatives, the author proposes new, non-intrusive methods that combine low memory demand with high computational efficiency. The new formulation of the first-order second-moment method requires only one additional function evaluation in total, regardless of the problem size. For the second-order fourth-moment method, the number of function evaluations scales linearly with the number of random variables, which is still one order lower than existing, non-intrusive approaches. Besides, a new probabilistic method tailored specifically to discontinuous response functions (which mainly occur for post-buckling optimization problems) is proposed.

In addition to the extra computational cost associated with the probabilistic framework, scattering variables may also affect the performance of finite element solvers. Considering anisotropic materials, such as wood or composites, the material orientation might be spatially random. For large-scale topology optimization, the Galerkin-based multigrid preconditioned conjugate gradient method has become the standard solver. However, the convergence rate is significantly reduced in the context of spatially varying material orientations. Without choosing the corresponding multigrid smoother properly, even divergence might occur. Therefore, the authors analyze different standard smoothing methods using numerical experiments and local Fourier analysis. A new smoothing algorithm called spatial Jacobi smoothing is proposed and validated using different examples.

Both improved probabilistic methods and spatial Jacobian smoothing, are applied to a robust topology optimization of a wooden three-point bending beam. The random fields of the uncertain material properties are measured and considered during the optimization process. Experimental validation shows improved robustness of the robust optimized design and good agreement between simulations and experiments.

Contents

Nomenclature

Abbreviations

FEM	finite element method
FOSM	first-order second-moment method
SOFM	second-order fourth-moment method
PSF	principal sensitivity first-order second-moment method
psFOSM	principal sensitivity first-order second-moment method
nPSF	numeric principal sensitivity first-order second-moment method
aPSF	analytic principal sensitivity first-order second-moment method
MMA	method of moving asymptotes
GCMMA	globally convergent method of moving asymptotes
psSOFM	principal sensitivity second-order fourth-moment method
SIMP	solid isotropic material with penalization
OLE	optimal linear estimator
EOLE	expansion optimal linear estimator
CFRP	carbon fiber reinforced polymer
GMRES	generalized minimal residual method
pcg	preconditioned conjugate gradients
LFA	local Fourier analysis
SOR	successive overrelaxation
PDE	partial differential equation
CPU	central processing unit
RDO	robust design optimization
dofs	degrees of freedom

Symbols

bold	matrix/vector
$\frac{df}{dy}$	total derivative of f w.r.t. y, including implicit relations
$\frac{\partial f}{\partial y}$	partial derivative of f w.r.t. y, including only explicit relations
$\boldsymbol{K}$	assembled stiffness matrix
$\boldsymbol{M}$	assembled mass matrix
$\boldsymbol{R}$	residual of finite element equation
$\boldsymbol{u}$	deformation vector
$\boldsymbol{\lambda}$	adjoint vector or Lagrange multiplier
$\boldsymbol{f}$	force vector

$\boldsymbol{x}$	vector of random variables
$\boldsymbol{z}_i$	position vector of element or node i
$\boldsymbol{y}$	vector of design variables
$\boldsymbol{\Sigma_x}$ or $\boldsymbol{Cov}$	covariance matrix of random variables $\boldsymbol{x}$
$\boldsymbol{V_x}$, $\boldsymbol{W_x}$	matrices of eigenvectors and eigenvalues of $\boldsymbol{\Sigma_x}$
$\mathrm{cov}(x_i, x_j)$	covariance between variables x_i and x_j
$H_x(\boldsymbol{z})$	continuous random field of variable x with position $\boldsymbol{z}$
$\boldsymbol{\mu_x}$	mean value of vector of random variables $\boldsymbol{x}$
μ_i	mean value random variable x_i
σ_{xi}	standard deviation of i-th entry of $\boldsymbol{x}$
$p(\boldsymbol{x})$	probability density function of variables $\boldsymbol{x}$
σ_f	standard deviation of response f
μ_f	mean value of vector of response f
κ	weighting factor
ϵ	scaling factor
V_0	volume of design space in topology optimization
l_c	correlation length
ρ	pseudo-density in topology optimization
ω_i	i-th eigenfrequency
$\boldsymbol{\varphi}_i$	i-th eigenvector
ω	damping factor for several multigrid smoothing methods
$\boldsymbol{\Theta}_j$	j-th error frequency
$\boldsymbol{v}$	error of the deformation vector
$\boldsymbol{S}(\Theta_j)$	amplification matrix function
μ_{loc}	smoothing factor
η	rotation angle for rotation around x-axis or threshold parameter in projection function
φ	rotation angle for rotation around z-axis
θ	rotation angle for rotation around y-axis
β	steepness parameter of projection function
$\boldsymbol{D}$	diagonal part of stiffness matrix
$\boldsymbol{L}$	strictly lower diagonal part of stiffness matrix
$\boldsymbol{D}_p$	block-diagonal part of stiffness matrix with couplings between dofs of same node
$\boldsymbol{L}_p$	strictly lower block-diagonal part of stiffness matrix with couplings between dofs of same node
$\boldsymbol{D}_L/P$	block-diagonal part of stiffness matrix with couplings between dofs of same line/plane
$\boldsymbol{L}_L/P$	strictly lower block-diagonal part of stiffness matrix with couplings between dofs of same line/plane
Θ	vector containing all rotation angles
E_i	Young's modulus of element i
Δy	difference related to variable y

Chapter 1

Introduction

Improving a given design has always been a fundamental part of engineering. Besides many part-specific objectives and constraints, huge efforts are taken to generally reduce the cost of the final product and the environmental impact.

Especially in the aerospace industry, the cost and the environmental impact are often driven by the weight of the product. The maximum payload of a rocket is strictly limited by the maximum available fuel and the engine power. Since rockets either can only be used once, or it takes huge effort to prepare the rocket for another launch, exploiting the maximum possible payload is crucial. According to the company SpaceX, the cost for launching a Falcon 9 rocket with up to 5500 kg payload to the geostationary transfer orbit is approximately 70 million US dollars [1], which leads to a cost of 11,000 dollars per kilogram. Hence, reducing the weight of the payload or increasing the maximum payload by reducing the weight of the rocket is economically attractive.

In air transportation, the cost per kilogram is much lower, with values around 4euros per kg [2] for a transport from China to the USA. Hence, reducing the weight of the payload is less attractive. However, in contrast to a rocket, an airplane is used many times without significant launch preparation cost. In consequence, it is highly economical to reduce the mass of the airplane itself, such that either the total transportation weight is increased or the fuel consumption is decreased over the whole lifetime.

In classical engineering, designs are manually generated using basic knowledge first and simulated afterwards. If the simulation reveals that the design has some flaws, it is sent back to the designer for improvement. This framework usually takes much time and only explores a highly limited part of the design space. In difference, structural optimization approaches directly integrate the simulation into an automated design process. Using gradient-based mathematical programming approaches, (local) optimal designs regarding given conditions are found after few automated design iterations. The generated designs are finally interpreted by the engineer and modified to meet the company's expectations. By application of gradient-based optimization methods to several parts of the Airbus A380 airplane, the overall weight has been reduced by 700 kg compared to the final design generated with classical engineering methods [3].

With the rise of metal 3D printing, also highly complex 3D designs became manufacturable, leading to a higher potential of structural optimization. Nowadays, several topology-optimized and 3D-printed parts have been authorized for the aircraft industry. For example, the topology optimization-inspired bracket shown in [4] has been mounted on the Airbus 350, leading to a

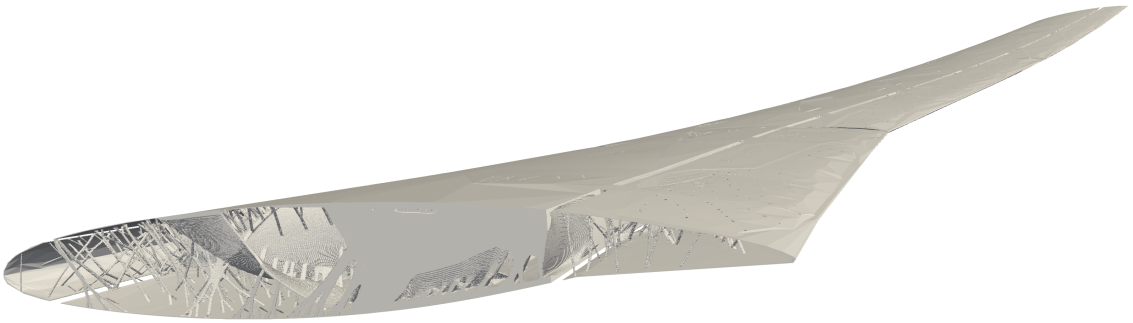


Figure 1.1: Topology optimization of a whole aircraft wing in ultra high resolution. The image is generated from the data provided in [7].

weight reduction of 40% compared to the base design [5]. Additionally, advances in algorithms for solving large systems of equations as well as increasing computing power enabled to optimize large-scale problems with billions of degrees of freedom. In consequence, it is possible to optimize a whole wing structure with a minimal feature size of less than 1 cm such as presented in [6] (see figure 1.1), reducing the overall weight by up to 15% compared to the reference design. However, the computational cost is still a challenge, especially in the case of spatially varying anisotropic materials, where the highly efficient system of equation solvers suffer and the stiffness matrix has to be recomputed for every finite element in every design iteration. Additionally, in practice, the real conditions slightly differ from the conditions assumed during the design process. For example, geometry deviations might be caused by manufacturing tolerances, load directions change due to imperfect mounting of a component and boundary conditions as well as load introduction are only modeled in an idealized way. This becomes crucial if the design is generated using structural optimization tools, since optimization algorithms only consider the given conditions and exploit all flaws in the problem formulation. For example the design shown in figure 1.1 is highly tailored to bending loads in vertical direction, and hence, might suffer from unexpected torsion loads. In classical engineering, safety factors are applied to address uncertain and unexpected effects. However, the safety factors lead to unnecessary conservative designs on the one side, and might not improve safety for specific situations on the other side [8].

As an alternative, probabilistic methods may be applied to statistically consider uncertainties that occur in real applications. Thereby, the parts are designed with respect to a specific probability of exceeding allowed limits. By considering uncertainties already during the optimization procedure, the final designs adapt to possible, unexpected situations and therefore add effective structures. In consequence, the optimizer addresses possible problems directly instead of just adding lots material. However, most methods for optimization under uncertainties suffer from high computational cost and therefore, this is an open challenge for research.

1.1 State of the art

The current dissertation builds up on developments of structural optimization, especially topology optimization and shape optimization, as well as optimization under uncertainty. Especially in the topology optimization of 3D structures, the problem size becomes large and hence, advanced solvers for the occurring systems of equations are required. There, the latest developments of the multigrid method are relevant.

1.1.1 Structural optimization

Structural optimization is usually divided into the fields parameter optimization, sizing optimization, topology optimization and shape optimization [9, 10]. In general, parameter optimization considers a limited set of arbitrary design parameters, such as diameters, distances and laminate plies. In difference, sizing optimization usually considers a high dimensional set of size parameters, such as the thickness of shell elements, which do not alter the finite element mesh. These two types are not considered explicitly in the current dissertation, although the presented methods are applicable to these problems.

The density-based topology optimization is based on using the densities of finite elements as design variables, which are modified using a mathematical optimization algorithm [11]. Since only gradient-based optimization approaches are suitable for such problems, usually different variants of the method of moving asymptotes [12, 13], convex optimization [14] or heuristic gradient-informed methods (e.g., evolutionary topology optimization [15]) are used as optimization algorithms. To avoid numerical instabilities and to enforce mesh independence at a given length scale, density filters [16, 17, 18, 19] as well as projection methods [20, 21, 22] have been proposed. During the last decades, topology optimization formulations have been developed and improved for several problems, such as optimization of geometrically nonlinear problems [23, 24], eigenvalues [25, 26], stresses [27, 28] and contact [29]. Additionally, it has been seldomly applied to natural materials like wood [30]. Besides the density-based topology optimization, also other topology optimization parameterizations exist, such as the level-set topology optimization [31] and feature mapping methods [32]. These approaches became more popular during the last years but are not considered in this work.

In difference, shape optimization considers the outer shape as design variables. In general, three major groups of shape optimization exist. The parameter-based approaches use CAD parameters (see, e.g., [33]) or spline parameters (see e.g., [34]) as design variables, whereas evolutionary algorithms start from a given design and locally increase the structural thickness at highly stressed regions. In difference, parameter-free approaches start from a finite element mesh and optimize the nodal coordinates directly [10]. In the current contribution, only the latter is considered. Early publications on gradient-based optimization of nonlinear structures are found in [34, 35, 36, 37]. There, nodal sensitivities are already computed using the adjoint method.

Using the nodal coordinates directly as design variables, numerical instabilities occur, which are caused by poor FE-approximations of distorted meshes and a missing length scale [38]. There are two major groups of approaches that address this challenge: the natural variable approach and filters based on convolution integrals. In the natural variable approach (see, e.g., [39, 40]) artificial loads are used as design variables. The loads are applied to the mechanical system and the resulting

deformation is used as the shape update. In difference, the filter based approach directly considers shape updates, which are filtered using a low pass filter known from image processing or a physically motivated filter operation [41]. In [42], both approaches are combined. There, the update of the surface nodes is done using the filter based approach. Afterwards, the inner nodes are updated using a natural variable approach by application of the surface node update as a forced displacement field. A similar approach is applied to tangential shape updates of shell models in [43]. More recent publications focus on an approach called vertex morphing, which builds on filter-based regularization [44]. Additionally, the development of improved optimization methods is part of recent research [45].

Besides the classic optimization problems, also combinations of different problems are part of current research. In [46], topology optimization and shape optimization are combined to find designs similar to classic topology optimization, which have smooth surfaces caused by the shape optimization. Alternatively, shape optimization of shell structures is combined with sizing optimization in [47] to increase the design freedom.

In the context of composite structures, different formulations exist to combine the optimization of the (spatial) composite orientation with optimization of the general shape of the design. Considering shell structures, the discrete material optimization is a special form of multi-material topology optimization. There, different candidate orientations are given for every finite element. Using a multi material topology optimization framework, the best candidate orientations and the corresponding element densities are computed [48]. In the discrete material and thickness optimization framework [49], also the thickness of each laminate layer is optimized.

For continuum models, the topology optimization with smoothly varying fiber orientations [50] considers two sets of design variables for every element: the densities and the orientation angles. Thereby, an optimized topology with corresponding material orientations is found. Alternatively, a representation based on splines is used in [51].

1.1.2 Large scale structural optimization

Especially in the topology optimization of big three-dimensional structures, a large number of finite elements is required to obtain designs with fine details (see e.g., wing in [6], bridge in [52]). Besides, also shape optimization problems might become large if either fine details or a high solution accuracy are required. Using direct solvers, the solution of the finite element equation becomes infeasible due to high computation time and high memory demands [53]. In this case, the cost of the system solve is dominant compared to all other parts of the problem solution (i.e. optimization step, pre-processing, post-processing). Therefore, parallel computing in combination with domain decomposition and iterative solvers has been used in the early days [53]. Different approaches aim at reducing the computational cost of iterative solvers, mainly the conjugate gradient method or the generalized minimal residual method, by recycling of Krylov subspaces [54] or by reusing preconditioners [55]. However, the methods based on standard preconditioners and recycling still have high limitations regarding scalability to large problems [56].

These limitations vanish by using the multigrid method as a preconditioner to the iterative solver (e.g., conjugate gradients). The multigrid method is an iterative solver based on successive solutions of the partial differential equation on coarser grids. The coarser grids might be defined geometrically (geometric multigrid method) or algebraically, using only the corresponding stiffness matrix of the fine grid (algebraic multigrid) [57]. The multigrid method is known in mathematics for many years [58] and has already been applied to topology optimization early [59]. However, it was unnoticed in

engineering until the educational paper by Amir et al. [60], where the geometric multigrid method is applied to the educational 88-line code [61]. Since then, the (geometric) multigrid method has been widely applied, for example, using the open source code PETSc [62], or to shell structures [63]. Besides, the topology optimization does not require the state equations to be solved accurately. It is only required that the resulting gradient roughly points into the correct direction, which is the case when using the geometric multigrid method [64]. Hence, Amir proposes a one shot procedure, where only one solver iteration is applied in each optimization iteration, leading to significantly reduced computation time [65]. However, it is restricted to linear elasticity.

Although the geometric multigrid method has been shown to be highly efficient for topology optimization problems, there still are remaining challenges. As shown in [60], the geometric multigrid method suffers from high contrasts in a design. Additionally, it is known that also anisotropies in the underlying partial differential equation, e.g., caused by anisotropic materials or high-aspect-ratio elements, worsen the convergence speed [57]. An alternative to this issue is to use the algebraic multigrid instead. It is shown that the algebraic multigrid framework reduces the overall computation time for 2D problems in the context of high contrasts or high aspect ratios, however, this does not hold for 3D problems [66].

By using efficient solvers for the system of equations, the computational cost of other parts of the optimization framework becomes more relevant. Therefore, a parallel formulation of the method of moving asymptotes is presented in [67] and general code improvements are proposed in [68], leading to a topology optimization framework that can be scaled to problems with more than one billion finite elements [52].

1.1.3 Optimization under uncertainty

Ways to consider uncertainties in a design process are often categorized based on the knowledge given for the uncertain variables. For example, the idea of an uncertainty triangle is presented in [69], where the author proposes to distinguish whether probabilistic data, partial knowledge or variable bounds are given for the uncertain variables. Depending on the given information, the author proposes to use either probabilistic methods, worst-case approximations or Fuzzy sets. An alternative categorization is the distinction between aleatoric and epistemic uncertainties, where epistemic uncertainties correspond to missing information or model errors whereas aleatoric uncertainties are caused by physical random effects (see, e.g., [70]). A combination of both is sometimes called polymorphic uncertainty [71]. In the design optimization, epistemic uncertainties are usually considered using non-probabilistic methods, such as worst-case approximations [72, 73, 74], fuzzy logic [75, 76] or interval arithmetic [77, 78]. In difference, aleatoric uncertainties are usually considered using probabilistic methods, such as Monte-Carlo experiments. In the current dissertation only aleatoric uncertainties are considered. However, the presented approaches may also be extended to epistemic uncertainties by adaptation to convex anti-optimization methods [79].

Overviews for optimization under stochastic uncertainties are given in [80, 81, 82]. Besides the worst-case approach, there are two major groups of optimization formulations. In the reliability-based design optimization (RBDO, review in [83]), the probability of failure (i.e. value exceeding allowed value) either is restricted by a constraint or is used as the objective function. In difference, the robust design optimization (RDO, review in [84]) is based on the Taguchi method [85], where the noise should be small compared to the mean value. Therefore, usually a weighted combination of mean value and the standard deviation is used as the robust objective function. Although the formulations are quite different, it has been shown that the resulting designs are sometimes

similar [86]. Especially it can be shown that the formulations are equal if a specific weighting factor is chosen for the robust design optimization. In the current publication the focus is set on robust design optimization. The main component of robust design optimization is to compute the stochastic moments (usually mean and standard deviation) and the corresponding gradients related to the objective function. Therefore, the stochastic integrals

$$\mu_f = \int_{-\infty}^{\infty} f(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{1.1}$$

$$\sigma_f^2 = \int_{-\infty}^{\infty} (f(\boldsymbol{x}) - \mu_f)^2 p(\boldsymbol{x}) d\boldsymbol{x} \tag{1.2}$$

with the objective function f and the probability density function p have to be solved. Since the objective function usually is highly nonlinear and probably not explicitly given, the integrals are hard to compute. Therefore, most probabilistic methods aim at approximating the stochastic moments.

The Monte Carlo method is known in probabilistics for many years (see, e.g., [87]) and has been applied to topology optimization with uncertainties of the geometry in [88, 89]. It is known to provide accurate results, if a sufficient sample size is given. However, it requires to evaluate the full objective function for every sample. Considering thousands to millions of samples, the computational cost is infeasible for structural optimization. Therefore, the number of samples is usually reduced during the optimization process leading to less accurate approximations. As an alternative, the Monte-Carlo method is often boosted by reduced order models [90] or different surrogate models, such as hybrid models [91], Kriging models [92], neural networks [93, 94] and polynomial chaos expansion [95, 96, 97]. However, the costly training (generating training data and parameter training) of the surrogate model must be done in every optimization iteration leading to bad scalability. For example, the minimal number of function evaluations for a first-order polynomial chaos expansion depends on the sampling method and scales at least linearly with the number of random variables, while higher order models scale even with a higher order. Therefore, the polynomial chaos expansion is enhanced in [98] using gradient information such that higher order approximations are more efficient.

As an alternative to the (enhanced) Monte Carlo method, stochastic gradient approaches have been applied to several optimization problems [99, 100]. Like the Monte Carlo method, it uses random samples during the optimization, but the number of samples is extremely low and different samples are generated every optimization iteration. In consequence, the computational cost per iteration is rather low while the approximation of moments is poor. Due to the generation of new samples every iteration, the convergence is less smooth leading to many optimization iterations, especially in the case of many random variables [101]. Therefore, Grieshammer et al. present the continuous stochastic gradient method, where the evaluations of previous iterations are used to improve the current approximation of the mean [101]. This approach is applied to different examples [102, 103], but still lacks scalability.

Alternatively, the integral equations (1.1) and (1.2) can be solved using quadrature rules, which is sometimes also referred as stochastic collocation method [104, 105, 106, 107]. Similar to the polynomial chaos expansion, the required number of function evaluations depends on the sampling method and hence, requires a similar computation time.

Another group of methods is based on Taylor series expansions instead of sampling based approaches. In the perturbation method, the mean and the covariance matrix of the state, e.g., the deformation field, is computed. Afterwards, the stochastic moments of the objective function are

computed from the state. Therefore, the state is approximated by a Taylor series with respect to the random parameters. Inserting the Taylor series into (1.1) and (1.2) leads to analytic expressions depending on the sensitivities of the state and the stochastic moments of the random variables. This approach has been widely applied to optimization of ground structures [108, 109] and to topology optimization [110, 111, 112]. However, computing all required derivatives of the state for the first-order perturbation approach requires one adjoint system per random variable leading to a lack of scalability. Higher order approaches are even more challenging.

In contrast, the methods of moments apply the Taylor series directly to the objective function. The first-order second-moment method (FOSM) is based on a first-order Taylor series approximation and has been applied to many examples, such as ground structures [113], topology optimization [114, 115, 116], sizing optimization [117], shape optimization [118, 119] and process optimization [120]. Independent of the number of random variables, it requires only two adjoint systems, and hence is highly scalable [115]. In many cases, it has the same accuracy as the first-order perturbation approach, but requires significantly less computation time. Besides, there is also a non-intrusive version [121], which however scales linearly with the number and hence, is not scalable. Unfortunately, many objective functions show a higher order relationship between objective and random variables, such that all methods based on first-order polynomials (e.g., FOSM, first-order perturbation, surrogates of order 1) suffer from inaccuracies. In difference, the second-order method of moments, also referred to as second-order fourth-moment method [114] or second-order perturbation approach [122], shows good accuracy for several examples [122, 114, 123]. On the contrary, the number of adjoint systems scales at least linearly with the number of random variables, whereas the number of function evaluations scales even quadratically for non-intrusive variants [121].

1.2 Objective, contributions and outline of the thesis

As mentioned previously, many probabilistic approaches suffer from the curse of dimensionality, meaning that the required number of systems of equations somehow scales with the number of random variables. This is either caused by the costly training of surrogates in every design iteration or the required computation of sensitivities of vector-valued functions (e.g., perturbation approaches). Other approaches suffer from missing flexibility, since the approaches are highly intrusive and problem specific (e.g., method of moments). In consequence, these formulations cannot be simply implemented for commercial codes and hence, are not well suited for industrial applications. Besides, many formulations require the computation of many matrix-valued partial derivatives, which leads to high memory requirements. The current dissertation addresses these challenges and proposes two new formulations for the methods of moments. In difference to the known formulations, the newly presented principal-sensitivity first-order second-moment method (psFOSM) is non-intrusive and requires only one additional function evaluation, independent of the number of variables, while the accuracy is equal to the first-order second-moment and similar to the first-order perturbation approach. The newly presented non-intrusive principal-sensitivity second-order fourth-moment method (psSOFM) is more accurate than the first-order version, but requires four additional function evaluations per random variable, which, however, can be processed mostly in parallel. Both methods do not increase the memory demands compared to a deterministic optimization, such that psFOSM is truly scalable, while psSOFM is at least more scalable than alternative approaches with similar accuracy. However, the author observed poor performance of the proposed methods for several geometric nonlinear examples. Under lead of Benedikt Kriegesmann, a new approach called “multimodal first-order second-moment” method is developed, which is specifically

tailored to these types of problems. The author contributed significant parts of the theoretical background and is responsible for the scalability of the proposed approach.

All new presented robust optimization approaches are applied to different examples of topology- and shape optimization for problems of linear and geometrically non-linear elasticity. While most examples are constructed, a robust optimization is applied to an experimentally motivated topology optimization example, considering uncertainties in the material data and the fiber orientation of a wooden base-material. Therefore, the uncertainties are measured using compression tests and image recognition to detect the probabilistic properties of the random variables. The final design is fabricated using a laser cutting machine and assessed experimentally afterwards. To exploit the potential of the fabrication method, a rather fine discretization is used, which requires the use of iterative system solvers. For large-scale problems, the conjugate gradient method preconditioned by the Galerkin-based multigrid method proved to be extremely efficient and scalable. However, it suffers from the (uncertain) anisotropies of the wooden base material. Therefore, a modified smoothing algorithm for the multigrid method is proposed, which improves the performance and enforces solver convergence for anisotropic materials.

The current thesis is organized as follows: In chapter 2, basic theory on the different optimization formulations, including optimization algorithms, parameterization, regularization and gradient computation is presented. Afterwards, in chapter 3, the basic equations of stochastics and random fields are given and the used probabilistic methods are derived. The following chapter 4 proposes new probabilistic methods. It contains two publications, where the new principal-sensitivity first-order second-moment method (publication 1, section 4.1) and the principal-sensitivity second-order fourth-moment method (publication 2, section 4.2) are derived as well as additional material deriving the multimodal first-order second-moment method. Chapter 5 addresses an experimentally motivated robust design optimization for a wooden three-point bending beam. Therefore, first, the use of the multigrid method for orthotropic materials is discussed and an alternative multigrid smoothing algorithm is proposed (publication 3, section 5.1). Using the proposed multigrid smoother as well as the previously developed probabilistic methods, a robust design optimization is applied and the final design is tested experimentally. Finally the thesis closes with a summary and outlook in chapter 6. The appendix contains the used geometrically non-linear finite element framework including the gradients with respect to nodal coordinates as well as some lengthy derivations.

Chapter 2

Basic theory on structural optimization in statics

In the current thesis, the optimization problems are formulated in the nested analysis and design (NAND) [124] formulation given by

$$\begin{aligned} &\min_{\boldsymbol{y}} f(\boldsymbol{y}, \boldsymbol{u}(\boldsymbol{y})) \\ &\text{s.t.} \\ &\boldsymbol{y}_l \leq \boldsymbol{y} \leq \boldsymbol{y}_u \\ &\boldsymbol{c}(\boldsymbol{y}, \boldsymbol{u}(\boldsymbol{y})) \leq \boldsymbol{0} \\ &(\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{0}) \end{aligned} \quad , \tag{2.1}$$

where the objective function f depends on the state $\boldsymbol{u}$ and the design variables $\boldsymbol{y}$. The computation of the state is nested into the function evaluation using the finite element analysis, such that the implicit equality condition $(\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{0})$ is automatically fulfilled. Additionally, the design variables are bounded by the lower bound $\boldsymbol{y}_l$ and the upper bound $\boldsymbol{y}_u$ as well as by the vector-valued inequality constraint $\boldsymbol{c}(\boldsymbol{y}, \boldsymbol{u}(\boldsymbol{y})) \leq 0$. Considering this optimization problem, a solution $\boldsymbol{y}^*$ is a local optimum if the Karush-Kuhn-Tucker conditions (KKT-conditions) given by

$$\begin{aligned} \frac{df}{d\boldsymbol{y}} + \boldsymbol{\lambda}\frac{d\boldsymbol{c}}{d\boldsymbol{y}} - \sum_i \lambda_{l,i} + \sum_i \lambda_{u,i} &= \boldsymbol{0} \\ \boldsymbol{\lambda} &\geq \boldsymbol{0} \\ \boldsymbol{\lambda}_l &\geq \boldsymbol{0} \\ \boldsymbol{\lambda}_u &\geq \boldsymbol{0} \\ \boldsymbol{c}(\boldsymbol{y}^*, \boldsymbol{u}(\boldsymbol{y}^*)) &\leq \boldsymbol{0} \\ \boldsymbol{y}_l - \boldsymbol{y}^* &\leq \boldsymbol{0} \\ \boldsymbol{y}^* - \boldsymbol{y}_u &\leq \boldsymbol{0} \\ \boldsymbol{\lambda}\boldsymbol{c}(\boldsymbol{y}^*, \boldsymbol{u}(\boldsymbol{y}^*)) &= \boldsymbol{0} \\ \boldsymbol{\lambda}_l\,(\boldsymbol{y}_l - \boldsymbol{y}^*) &= \boldsymbol{0} \\ \boldsymbol{\lambda}_u\,(\boldsymbol{y}^* - \boldsymbol{y}_u) &= \boldsymbol{0} \end{aligned} \quad , \tag{2.2}$$

with the Lagrange multipliers $\boldsymbol{\lambda}, \boldsymbol{\lambda}_l, \boldsymbol{\lambda}_u$, are fulfilled. However, in practical application to high-dimensional topology optimization and shape optimization, the optimization algorithms find a nearly optimal solution very fast, whereas convergence to a solution fulfilling the Karush-Kuhn-Tucker conditions takes a huge amount of iterations. Therefore, it is common to use a stopping criterion instead of a true convergence criterion. In the current thesis, the optimization is usually stopped if the change of the function value between two subsequent iterations is less than a chosen threshold value.

2.1 Optimization algorithms

To find the local optimal value $\boldsymbol{y}^*$ of the optimization problem (2.1), different algorithms are available. Since the method of moving asymptotes (MMA, [12]) has been shown to be very efficient for structural optimization, it is applied in most considered examples of this thesis. Using the MMA, the objective and constraint functions of the original optimization problem are approximated by the functions

$$\tilde{f} = \sum_{i=1}^{N} \left(\frac{p_{i,f}}{u_i - y_i} + \frac{q_{i,f}}{y_i - l_i} \right) + r_f \tag{2.3}$$

$$\tilde{\boldsymbol{c}} = \sum_{i=1}^{N} \left(\frac{\boldsymbol{p}_{i,c}}{u_i - y_i} + \frac{\boldsymbol{q}_{i,c}}{y_i - l_i} \right) + \boldsymbol{r}_c \tag{2.4}$$

with

$$p_{i,f} = (u_i - y_i)^2 \left(1.001 \max(\frac{df}{dy_i}, 0) + 0.001 \max(-\frac{df}{dy_i}, 0) + \frac{10^{-5}}{y_{u,i} - y_{l,i}} \right) \tag{2.5}$$

$$q_{i,f} = (u_i - y_i)^2 \left(1.001 \max(-\frac{df}{dy_i}, 0) + 0.001 \max(\frac{df}{dy_i}, 0) + \frac{10^{-5}}{y_{u,i} - y_{l,i}} \right) \tag{2.6}$$

$$\boldsymbol{p}_{i,c} = (u_i - y_i)^2 \left(1.001 \max(\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + 0.001 \max(-\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + \frac{10^{-5}}{y_{u,i} - y_{l,i}} \right) \tag{2.7}$$

$$\boldsymbol{q}_{i,c} = (u_i - y_i)^2 \left(1.001 \max(-\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + 0.001 \max(\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + \frac{10^{-5}}{y_{u,i} - y_{l,i}} \right) \tag{2.8}$$

and

$$r_f = f(\boldsymbol{y}) - \sum_{i=1}^{N} \left(\frac{p_{i,f}}{u_i - y_i} + \frac{q_{i,f}}{y_i - l_i} \right) \tag{2.9}$$

$$\boldsymbol{r}_c = \boldsymbol{c}(\boldsymbol{y}) - \sum_{i=1}^{N} \left(\frac{\boldsymbol{p}_{i,c}}{u_i - y_i} + \frac{\boldsymbol{q}_{i,c}}{y_i - l_i} \right) \tag{2.10}$$

in every optimization iteration. The asymptote parameters u_i, l_i are usually updated in every iteration by a specific scheme, such that the optimizer shows the desired behavior. Using these approximations, the design update is computed using the optimal solution of the more simple

subproblem

$$\begin{aligned} &\min_{\boldsymbol{y},\boldsymbol{y}_+} \tilde{f}(\boldsymbol{y}) + \boldsymbol{\lambda}_+^T \boldsymbol{y}_+ + \frac{1}{2}\boldsymbol{y}_+^T \boldsymbol{y}_+ \\ &\text{s.t.} \\ &\tilde{\boldsymbol{c}}(\boldsymbol{y}) - \boldsymbol{y}_+ \leq \boldsymbol{0} \\ &\tilde{\boldsymbol{y}}_l \leq \boldsymbol{y} \leq \tilde{\boldsymbol{y}}_u \\ &\boldsymbol{y}_+ \geq \boldsymbol{0} \end{aligned} \quad , \tag{2.11}$$

where $\tilde{\boldsymbol{y}}_l$ and $\tilde{\boldsymbol{y}}_u$ are the move limits, which are also updated in every iteration. The vector-valued, penalty-like parameter $\boldsymbol{\lambda}_+$ is chosen by the user and usually set to a high value.

In some situations, the classic formulation of the MMA suffers from a missing convergence-control, probably leading even to divergence. This especially occurs if one of the functions is highly nonlinear, such as in the case of buckling-sensitive structures. For these situations, the globally convergent method of moving asymptotes (GCMMA, [13]) leads to better results. The basic ersatz-problem is equal to the classic MMA, however the factors $p_{i,f}, q_{i,f}, \boldsymbol{p}_{i,c}, \boldsymbol{q}_{i,c}$ are computed differently by

$$p_{i,f} = (u_i - y_i)^2 \left(1.001 \max(\frac{df}{dy_i}, 0) + 0.001 \max(-\frac{df}{dy_i}, 0) + \frac{\Delta_{MMA}}{y_{u,i} - y_{l,i}} \right) \tag{2.12}$$

$$q_{i,f} = (u_i - y_i)^2 \left(1.001 \max(-\frac{df}{dy_i}, 0) + 0.001 \max(\frac{df}{dy_i}, 0) + \frac{\Delta_{MMA}}{y_{u,i} - y_{l,i}} \right) \tag{2.13}$$

$$\boldsymbol{p}_{i,c} = (u_i - y_i)^2 \left(1.001 \max(\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + 0.001 \max(-\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + \frac{\Delta_{MMA}}{y_{u,i} - y_{l,i}} \right) \tag{2.14}$$

$$\boldsymbol{q}_{i,c} = (u_i - y_i)^2 \left(1.001 \max(-\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + 0.001 \max(\frac{d\boldsymbol{c}}{dy_i}, \boldsymbol{0}) + \frac{\Delta_{MMA}}{y_{u,i} - y_{l,i}} \right) \tag{2.15}$$

with the additional parameter Δ_{MMA} . During each optimization iteration, the parameter Δ_{MMA} is updated until the ersatz model is conservative meaning that $\tilde{f}(\boldsymbol{y} + \Delta\boldsymbol{y}) > f(\boldsymbol{y} + \Delta\boldsymbol{y})$ and $\tilde{\boldsymbol{c}}(\boldsymbol{y} + \Delta\boldsymbol{y}) > \boldsymbol{c}(\boldsymbol{y} + \Delta\boldsymbol{y})$. The design update is finally done using the solution of the MMA subproblem (2.11) with the updated parameter Δ_{MMA}.

2.2 Gradient computation

Using gradient-based optimization algorithms, computing accurate sensitivities becomes crucial. The simplest approach is to use the (forward) finite difference method, where the sensitivities are computed using

$$\frac{df}{dy_i} = \frac{f(y_i + \Delta y_i) - f(y_i)}{\Delta y_i} \quad , \tag{2.16}$$

where Δy_i represents the perturbation of the variable y_i. To achieve a higher accuracy, also higher order approaches, such as central differences, are available. However, for every variable, the function has to be evaluated at least once, leading to extreme computational cost for structural optimization.

Using the direct differentiation approach, the gradient of the objective/constraint function is computed from the gradient of the state $\boldsymbol{u}$ using the chain rule. A general derivative reads

$$\frac{df}{dy_i} = \frac{\partial f}{\partial y_i} + \frac{\partial f}{\partial \boldsymbol{u}} \frac{d\boldsymbol{u}}{dy_i} \tag{2.17}$$

and the gradient of the state is computed by

$$\frac{d\boldsymbol{u}}{dy_i} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}^{-1} \frac{\partial \boldsymbol{R}}{\partial y_i} \quad , \tag{2.18}$$

with the general residual $\boldsymbol{R}$. In the case of linear finite elements, the general residual reads $\boldsymbol{R} = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}$. Similar to finite differences, using the direct differentiation approach requires one system solve per variable, which makes this method unusable for structural optimization problems.

In difference, the adjoint method presented in [125] requires only one system of equations per function, independent of the number of variables. Therefore, the function is augmented by the scaled state equation, leading to

$$f = f + \boldsymbol{\lambda}^T \boldsymbol{R} \quad , \tag{2.19}$$

with the adjoint vector $\boldsymbol{\lambda}$ and the residual $\boldsymbol{R}$. Differentiation and reordering leads to

$$\frac{df}{dy_i} = \frac{\partial f}{\partial y_i} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial y_i} + \left(\boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} + \frac{\partial f}{\partial \boldsymbol{u}} \right)^T \frac{d\boldsymbol{u}}{dy_i} \quad . \tag{2.20}$$

Setting $\boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} + \frac{\partial f}{\partial \boldsymbol{u}} = \boldsymbol{0}$ results in

$$\frac{df}{dy_i} = \frac{\partial f}{\partial y_i} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial y_i} \tag{2.21}$$

$$\boldsymbol{0} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} + \frac{\partial f}{\partial \boldsymbol{u}} \quad . \tag{2.22}$$

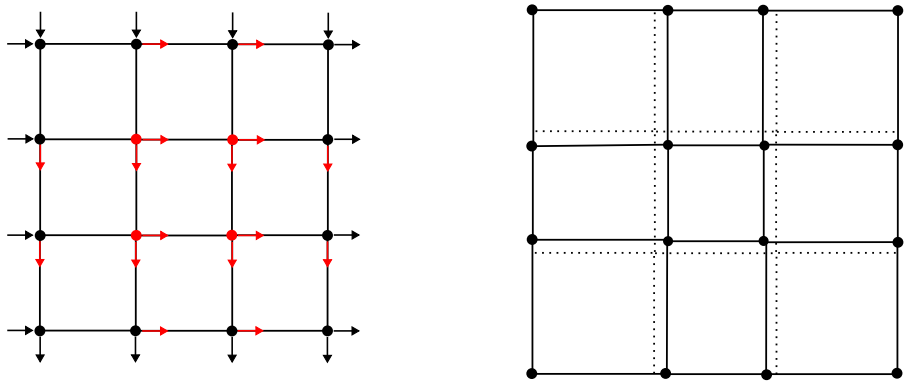
The adjoint system $\boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} + \frac{\partial f}{\partial \boldsymbol{u}} = \boldsymbol{0}$ is independent from the parameterization and hence has to be solved only once.

2.3 Shape optimization

The shape optimization problems of the current thesis are mainly built on the approach by Le et al. [42], where the nodal coordinates are considered to be variable. As depicted in figure 2.1a, the nodes are classified as internal nodes and surface nodes. The internal nodes do not describe the shape of the structure and hence, do not have physical influence on the structural performance. The same holds for the degrees of freedom of the surface nodes pointing into tangential direction. Therefore, only the normal movements of the surface nodes $\boldsymbol{y}$ are considered as design variables, while the inner nodes and tangential movements are updated respectively, such that the mesh quality is acceptable. In order to encounter numerical instabilities due to errors of the finite element method, the movement of the surface nodes is filtered using the vertex morphing approach reviewed in [44]. In the current thesis, a first-order discretization and a first-order filter function is considered, such that the true nodal movement $\Delta \boldsymbol{z}_i$ of node i is computed by

$$\Delta \boldsymbol{z}_i = \frac{\sum_j \boldsymbol{y}_j \cdot \max(R - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|, 0)}{\sum_j \max(R - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|, 0)} \quad , \tag{2.23}$$

with the filter radius R, the position $\boldsymbol{z}_i$ of node i and the design variables $\boldsymbol{y}_i$ related to node i. The movement of the inner nodes is computed by the concept of Laplacian smoothing, as depicted in



(a) Classification of nodes as interior (grey) or surface nodes (black). The normal movements of the surface nodes (black arrows) are considered as design variables, while the remaining movements (grey arrows) are updated to conserve mesh quality.

(b) Smoothing of the inner nodes by application of Laplacian smoothing. The solid lines represent the mesh before smoothing, the dotted lines after smoothing.

Figure 2.1: Parameterization and mesh regularization in parameter-free shape optimization.

figure 2.1b . There, the movement of the surface nodes $\Delta \boldsymbol{x}_{surf}$ is applied as forced displacements to an auxiliary elasticity problem. The resulting displacements are used as movements of the inner nodes $\Delta \boldsymbol{z}_{in}$, leading to the system of equations

$$\boldsymbol{K}_{in,in} \Delta \boldsymbol{z}_{in} = \boldsymbol{K}_{in,surf} \Delta \boldsymbol{z}_{surf} \quad , \tag{2.24}$$

with the partial stiffness matrices $\boldsymbol{K}_{in,in}$ (only inner nodes) and $\boldsymbol{K}_{in,surf}$ (relation surface nodes to inner nodes).

2.4 Density-based topology optimization

In the density-based topology optimization, element-wise pseudo densities $\rho_i \in [0,1]$ are chosen as design variables, where $\rho_i = 1$ is interpreted as material and $\rho_i = 0$ is interpreted as void. These densities define the Young's modulus of each element. In consequence, the full optimization problem reads

$$\begin{aligned} &\min_{\boldsymbol{\rho}} f(\boldsymbol{\rho}, \boldsymbol{u}(\boldsymbol{\rho})) \\ &\text{s.t.} \\ &0 \leq \rho_i \leq 1 \quad \forall i \quad , \\ &\boldsymbol{c}(\boldsymbol{\rho}, \boldsymbol{u}(\boldsymbol{\rho})) \leq \boldsymbol{0} \\ &(\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{\rho}) = \boldsymbol{0}) \end{aligned} \tag{2.25}$$

with the vector $\boldsymbol{\rho}$ containing all elemental densities ρ_i. To avoid intermediate densities, usually a progressive function is chosen for the relationship between pseudo density and Young's modulus.

Commonly used approaches are the "Solid Isotropic Material with Penalization" (SIMP) [126] and the "Rational Approximation of Material Properties" (RAMP) [127] which are both motivated from the Hashin-Shtrikman bounds. Using the modified solid isotropic material with interpolation (SIMP) model [21], the element Young's modulus E_i of element i is given by

$$E_i = E_{min} + (E_0 - E_{min})\rho_i^p \quad , \tag{2.26}$$

where E_0 represents the Young's modulus of the material, E_{min} is the artificial Young's modulus of void, usually set to $E_{min} = 10^{-9}E_0$ and p the penalization parameter. Usually, the penalization parameter is either set to $p = 3$ [126] or a continuation scheme is applied [128]. Without further regularization, the results are highly mesh dependent and numeric effects called "checkerboarding" occur. The most common solution is to use filters, such as the density filter [16, 21] and the Helmholtz filter [17].

2.4.1 Filtering

Using the density filter, the filtered densities $\tilde{\rho}_j$ are computed using a weighted average of design-densities ρ_i by

$$\tilde{\rho}_j = \frac{\sum_i w\left(\boldsymbol{z}_i\right) v_i \rho_i}{\sum_i w\left(\boldsymbol{z}_i\right) v_i} \quad . \tag{2.27}$$

Here, v_i represents the volume of element i and the function $w\left(\boldsymbol{z}_i\right)$ is a linearly decaying weighting function given by

$$w\left(\boldsymbol{x}_i\right) = \max(R - ||\boldsymbol{z}_i - \boldsymbol{z}_j||, 0) \quad , \tag{2.28}$$

where R represents the user chosen filter radius and $\boldsymbol{z}_{i/j}$ is the position vector of element i or j. More commonly, this operation is described using the design vector $\boldsymbol{\rho}$ and the vector of filtered densities $\tilde{\boldsymbol{\rho}}$ by

$$\tilde{\boldsymbol{\rho}} = \boldsymbol{T}_{f,d}\boldsymbol{\rho} \quad , \tag{2.29}$$

where the filter matrix $\boldsymbol{T}_{f,d}$ is a sparse matrix, containing the coefficients of (2.27).

In the context of large models, fine discretizations or large filter radii, the filter matrix $\boldsymbol{T}_f$ contains many non-zero entries leading to a high setup time, huge memory consumption and high operation time.

In difference, the Helmholtz filter is computationally efficient for large problems. It minimizes the potential

$$\Pi(\tilde{\rho}) = \frac{1}{2}\int_\Omega r^2 \left|\nabla\tilde{\rho}\right|^2 + (\rho - \tilde{\rho})^2 d\Omega \tag{2.30}$$

within the domain Ω by formulating the minimum of the problem as the solution of the partial differential equation

$$-r^2\nabla^2\tilde{\rho} + \tilde{\rho} = \rho \tag{2.31}$$

with the Neumann boundary condition

$$\frac{\partial\rho}{\partial\boldsymbol{n}} = 0 \quad . \tag{2.32}$$

Here, ρ represents the continuous design density, $\tilde{\rho}$ the continuous filtered density and r the filter radius. The partial differential equation is solved using the finite element method on the same finite element mesh as the elasticity problem. Hence, the problem is written as

$$\boldsymbol{K}_f\tilde{\boldsymbol{\rho}}_n = \boldsymbol{T}_f\boldsymbol{\rho} \tag{2.33}$$

with the vector of filtered node densities $\tilde{\boldsymbol{\rho}}_n$ and the element wise design densities $\boldsymbol{\rho}$. The filter matrices $\boldsymbol{K}_f$ and $\boldsymbol{T}_f$ are computed elementwise by

$$\boldsymbol{K}_{f,i} = \int_{\Omega_i} \nabla \boldsymbol{N}_e^T r^2 \nabla \boldsymbol{N}_e + \boldsymbol{N}_e^T \boldsymbol{N}_e d\Omega \tag{2.34}$$

$$\boldsymbol{T}_{f,i} = \int_{\Omega_i} \boldsymbol{N}_e^T d\Omega \tag{2.35}$$

with the matrices of the ansatz functions $\boldsymbol{N}_e$. The filtered element densities $\tilde{\boldsymbol{\rho}}$ are then computed by

$$\tilde{\boldsymbol{\rho}} = \boldsymbol{T}_f^T \cdot \tilde{\boldsymbol{\rho_n}} \tag{2.36}$$

which turns out to be the mean density in an element. As shown in [17], the Helmholtz filter is equivalent to the density filter, if $r = 2\sqrt{3}R$. Since the Helmholtz filter is based on the solution of a scalar partial differential equation, computation time and memory demands are independent of the filter radius. Besides, also the overall memory demands are extremely low due to the sparsity of the filter stiffness matrix $\boldsymbol{K}_f$. The system of equation (2.33) can be solved efficiently using iterative methods, leading to a low computation time.

To get a smooth transition zone between material and void at the design domain boundary, zero padding may be applied. For the Helmholtz-Filter, a very efficient formulation is given by the modified potential

$$\Pi_{aug}(\tilde{\rho}) = \frac{1}{2} \int_\Omega r^2 \left|\nabla \tilde{\rho}\right|^2 + (\rho - \tilde{\rho})^2 d\Omega + \int_\Gamma l_S \tilde{\rho}^2 d\Gamma \tag{2.37}$$

in [19]. There, a surface integral over the design domain surface Γ is added, penalizing material at the boundary of the design domain. The additional filter parameter l_S is chosen by the user and controls the effect of the zero padding. Using $l_S = 0$ is equal to no zero padding, when using $l_S = r$, material at the design domain boundary is treated equal to material inside the volume, and higher values penalize material at the design domain boundary even more. Minimization of the potential again leads to equation (2.31), but with the inhomogeneous boundary condition

$$\frac{\partial \rho}{\partial \boldsymbol{n}} = -l_S \tilde{\rho} \quad . \tag{2.38}$$

In the context of finite elements the problem is discretized to

$$\boldsymbol{K}_f \tilde{\boldsymbol{\rho}} + \boldsymbol{K}_{zp} \tilde{\boldsymbol{\rho}} = \boldsymbol{T}_f \boldsymbol{\rho} \tag{2.39}$$

where the matrices $\boldsymbol{K}_f$ and $\boldsymbol{T}_f$ are the same as in the classic Helmholtz filter, whereas the new filter matrix $\boldsymbol{K}_{zp}$ is computed element-wise by

$$\boldsymbol{K}_{zp,i} = \int_{\Gamma_i} \boldsymbol{N}_e^T \boldsymbol{N}_e d\Gamma \quad , \tag{2.40}$$

with the design domain surface of at element i Γ_i.

2.4.2 Projection and geometric uncertainty

To reduce intermediate densities or to enforce a specific length scale, the filtered densities $\tilde{\rho}_i$ are sometimes projected. In many cases (e.g. [20, 21]), the smooth heavyside projection given by

$$\hat{\rho}_i = \frac{\tanh(\beta\eta) + \tanh(\beta\tilde{\rho}_i - \beta\eta)}{\tanh(\beta\eta) + \tanh(\beta - \beta\eta)} \quad , \tag{2.41}$$

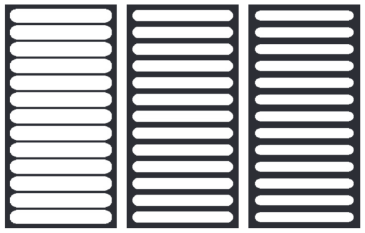


Figure 2.2: Effect of different projection thresholds. Using a higher threshold parameter (left), the whole design is eroded, using a lower parameter (right), the design is dilated compared to the neutral design (center). The image is taken from Schevenels et. al. in [88]

with the projection steepness parameter β and the projection threshold η, is used.

As proposed in [88], the projection function can also be used to represent geometric uncertainties. As depicted in figure 2.2, the design is eroded by an increased projection threshold and dilated by an decreased threshold respectively. However, uncertainties in the global threshold parameter only represent global imperfections. To consider local imperfections, the projection threshold is defined element-wise, such that

$$\hat{\rho}_i = \frac{\tanh(\beta\eta_i) + \tanh(\beta\tilde{\rho}_i - \beta\eta_i)}{\tanh(\beta\eta_i) + \tanh(\beta - \beta\eta_i)} \quad . \tag{2.42}$$

In this case, the uncertain threshold parameter characterized by a random field with a predefined correlation length, such that a minimal length scale of imperfections is ensured.

2.5 Robust design optimization

The robust design optimization builds on the basic structural optimization approaches, such as topology optimization and shape optimization. It uses the same parameterization (filters, projection, ...), analysis and boundary conditions. However, instead of the design response $f(\boldsymbol{y}, \boldsymbol{x})$ depending on the design variables $\boldsymbol{y}$ and uncertain variables $\boldsymbol{x}$, the robust objective

$$f_p = \mu_f(\boldsymbol{y}) + \kappa\sigma_f(\boldsymbol{y}) \tag{2.43}$$

is optimized. The design-dependent mean value μ_f and standard deviation σ_f are computed using probabilistic methods (see next section) based on the finite element simulation, while the weighting factor κ is chosen by the user. Usual values for κ are $\kappa \in [1, 3]$, which represent the single-sided 70% to 99% quantile values for many random distributions [129]. Besides, instead of the robust objective, also robust constraint functions may be defined, respectively. The general methods do not change, but no problems with probabilistic constraints are considered in the current thesis.

Chapter 3

Basics of Stochastics and Probabilistics

The basic equations and methodologies of the following stochastic basics are, for example, found in educational textbooks [130, 131]. For scalar random variables x, the statistical properties are usually quantified using stochastic moments (e.g. mean μ_x, standard deviation σ_x), quantile values (e.g., 5% quantile $x_{5\%}$) and the probability density function $p(x)$.

In the current thesis, mainly the first two statistical moments are considered, whereas all other statistical properties are only considered implicitly. The mean value μ_x (also referred to as expected value) of a variable x is computed by

$$\mu_x = E[x] = \int_{-\infty}^{\infty} x p(x) dx \quad , \tag{3.1}$$

with the general expectation value function $E[x]$. The variance σ_x^2 (squared standard deviation) is the second central stochastic moment, and hence, computed by

$$\sigma_x^2 = E[(x-\mu_x)^2] = \int_{-\infty}^{\infty} (x-\mu_x)^2 p(x) dx \quad . \tag{3.2}$$

If only empirical data is available, mean and variance are optimally approximated by

$$\mu_x = \frac{1}{n_{Data}} \sum_{i=1}^{n_{Data}} x_i \tag{3.3}$$

$$\sigma_x^2 = \frac{1}{n_{Data}-1} \sum_{i=1}^{n_{Data}} (x_i - \mu_x)^2 \quad , \tag{3.4}$$

with the number of samples n_{Data} and the i-th sample value x_i. For the more general problem of multiple random variables, the mean value vector $\boldsymbol{\mu_x}$ and the covariance matrix $\boldsymbol{\Sigma_x}$ are used. The entries of the covariance matrix are the covariances σ_{x_i,x_j}^2 of different variables x_i, x_j and the

variances σ_{xi}^2 such that

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \begin{pmatrix} \sigma_{x_1}^2 & \sigma_{x_1,x_2}^2 & \sigma_{x_1,x_3}^2 & \dots \\ \sigma_{x_1,x_2}^2 & \sigma_{x_2}^2 & \sigma_{x_2,x_3}^2 & \dots \\ \sigma_{x_1,x_3}^2 & \sigma_{x_2,x_3}^2 & \sigma_{x_3}^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.5}$$

with

$$\sigma_{x_i,x_j}^2 = E[(x_i - \mu_{x_i})(x_j - \mu_{x_j})] = \int_{-\infty}^{\infty} (x_i - \mu_{x_i})(x_j - \mu_{x_j}) p(\boldsymbol{x}) d\boldsymbol{x} \quad . \tag{3.6}$$

The non-diagonal entries are caused by correlation of the different random variables. Using the Pearson correlation measure $\mathbb{R}(x_i, x_j)$, the covariance is given by

$$\sigma_{x_i,x_j}^2 = \sigma_{x_i}\sigma_{x_j}\mathbb{R}(x_i, x_j) \quad . \tag{3.7}$$

In the case of spatially distributed uncertain variables, such as spatially varying material properties, the uncertainties correspond to continuous random fields $H_x(\boldsymbol{z})$ with the position $\boldsymbol{z}$ instead of discrete random variables. An overview of random fields is given in [132] and [133]. In this case, the stochastic properties are described by the mean value function $\mu_x(\boldsymbol{z})$ and the covariance function $\mathrm{cov}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j))$. Using the function-valued Pearson correlation coefficient $\mathbb{R}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j))$, the covariance function reads

$$\mathrm{cov}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j)) = \mathbb{R}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j))\sigma_x(\boldsymbol{z}_i)\sigma_x(\boldsymbol{z}_j) \quad . \tag{3.8}$$

Commonly used correlation functions are [134]

- exponential $\mathbb{R}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j)) = \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|}{l_c}\right)$
- squared exponential $\mathbb{R}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j)) = \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|}{l_c}^2\right)$
- γ-exponential $\mathbb{R}(x(\boldsymbol{z}_i), x(\boldsymbol{z}_j)) = \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|}{l_c}^\gamma\right)$

and their anisotropic variants, where $\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|}{l_c}$ is interchanged by $\frac{\|z_{1,i} - z_{1,j}\|}{l_{c,1}} + \frac{\|z_{2,i} - z_{2,j}\|}{l_{c,2}} + \frac{\|z_{3,i} - z_{3,j}\|}{l_{c,3}}$ with $\boldsymbol{z}_i = \begin{pmatrix} z_{1,i} \\ z_{2,i} \\ z_{3,i} \end{pmatrix}$ and the correlation length l_c. In the special case of a homogeneous random field, mean and variance are constant over the whole field and only the correlation function considers spatial effects.

3.1 Computation of the covariance function - semivariogram

In an experimental setup, the covariance function is computed using a semivariogram, as described in [135]. Under the assumption of homogenous random fields, the (spatial) distance $\Delta z_k = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$ and the squared value difference $\Delta x_k = (x_i - x_j)^2$ are computed for every pair k of samples i, j.

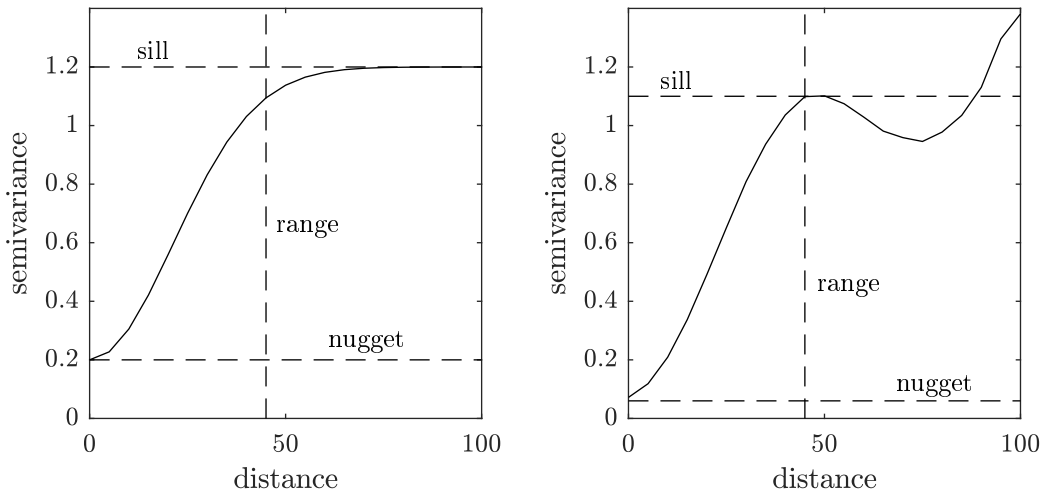


Figure 3.1: Ideal variogram of a square exponential random field considering infinite samples (left) and the variogram of a square exponential random field with limited samples (right).

All pairs are assigned to distance groups $\mathcal{K}_{\alpha,\beta} = [k : \alpha \leq \Delta z_k < \beta]$ and for each distance group, the semivariance $\gamma(\Delta z)$ is computed by

$$\gamma\left(\frac{\alpha+\beta}{2}\right) \approx \frac{1}{2N_{\alpha,\beta}} \sum_{k \in \mathcal{K}_{\alpha,\beta}} \Delta x_k \quad , \tag{3.9}$$

where $N_{\alpha,\beta}$ represents the number of elements of $\mathcal{K}_{\alpha,\beta}$. Plotting the semivariance over the distance leads to a semivariogram like shown in figure 3.1. If sufficient samples are available, the semivariance shows asymptotic behavior, where the supremum is called sill and the distance, at which the supremum is nearly reached, is called range. In practice, the semivariance is larger than zero for the zero-distance group. This value is called nugget and is caused by random effects of the experiment and measurement errors. Often, no sufficient number of samples is available. In this case, rather accurate values for range and sill are observed, but the semivariance shows wavy behavior for high distances. To compute the covariance function, a general guess about the type of correlation function (e.g. exponential) is made based on the semivariogram of the empirical data. Afterwards, the parameters of the covariance function are fitted, such that the difference between model semivariance and empiric semivariance are minimal.

3.2 Discretization of random fields

As mentioned before, random fields represent continuous spatial random variables. However, the simulation tools based on finite elements are based on discrete values. Therefore, the continuous random field $H_x(\boldsymbol{z})$ is discretized into many random variables x_i at the positions $\boldsymbol{z}_i$. The main approaches are discussed and compared in [133].

Using the midpoint method [136], the random variables x_i are defined element wise (with element i) and the corresponding values are taken from the continuous field at the element center $\boldsymbol{z}_i$, such that $x_i = H_x(\boldsymbol{z}_i)$ and hence, the covariance matrix is computed by evaluating the covariance function for the center positions. In consequence, the original random field is approximated by a piecewise constant function.

In difference, the more accurate integration point method [137] discretizes the random variables at the integration points corresponding to the integration of the finite element equation. Again, the values x_i at the Gauß-Point i are taken from the continuous random field such that $x_i = H_x(\boldsymbol{z}_i)$. If a sufficient number of integration points (i.e., random variables) is chosen for the integration of the correlation function, the random field is treated exactly. However, since the correlation functions are usually of high order (i.e., requiring many integration points), a huge amount of random variables (and following a huge covariance matrix) is required for exact integration.

As a compromise, the optimal linear estimator method (OLE) [138] approximates the true random field $H_x(\boldsymbol{z})$ as a piecewise linear function $\hat{H}_x(\boldsymbol{z})$ based on the random field values at different control points $\boldsymbol{z}_i$ as

$$H_x(\boldsymbol{z}) \approx \hat{H}_x(\boldsymbol{z}) = a(\boldsymbol{z}) + b^T(\boldsymbol{z}) \cdot \boldsymbol{\chi} \tag{3.10}$$

with $\boldsymbol{\chi} = [H_x(\boldsymbol{z}_1), H_x(\boldsymbol{z}_2), \dots]^T$. The control points can be chosen arbitrarily. The corresponding parameter functions $a(\boldsymbol{z})$ and $b(\boldsymbol{z})$ are chosen as the solution of the optimization problem

$$\begin{aligned} \forall \boldsymbol{z} \in \Omega \qquad & \min \quad \mathrm{var}[H_x(\boldsymbol{z}) - \hat{H}_x(\boldsymbol{z})] \\ & s.t. \quad E[H_x(\boldsymbol{z}) - \hat{H}_x(\boldsymbol{z})] = 0 \end{aligned} \tag{3.11}$$

with the considered domain Ω and the variance function var. The solution is known as

$$a(\boldsymbol{z}) = E[H_x(\boldsymbol{z})] - \mathrm{cov}[H_x(\boldsymbol{z}), \boldsymbol{\chi}] \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \mu_{\boldsymbol{\chi}} \tag{3.12}$$

$$b^T(\boldsymbol{z}) = \mathrm{cov}[H_x(\boldsymbol{z}), \boldsymbol{\chi}] \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \quad . \tag{3.13}$$

The piecewise linear approximation is afterwards considered analytically, which might be hard to implement in practice.

The different discretization methods are illustrated in figure 3.2. There, the same random field is discretized by the mentioned approaches for two different resolutions. From the figure it is observed that the approximating methods converge to the exact representation for a sufficiently fine mesh. This is usually the case for topology optimization problems, since there, a very fine discretization is already required to represent the geometric features. Hence, the simple midpoint method is a good choice for topology optimization problems. However, the fine discretization in combination with a significant correlation length leads to huge memory demands for storing the covariance matrix, also when using the midpoint method.

To reduce the memory demands for storing the covariance matrix, different approaches are available. First, the optimal linear estimator method may be used to reduce the number of random variables. Therefore, the approximated random field $\hat{H}_x(\boldsymbol{z})$ is evaluated at the element centers, while the control points are chosen on a coarse mesh. In this case, a random realization reads

$$x_i = \hat{H}_x(\boldsymbol{z}_i) = a(\boldsymbol{z}_i) + b^T(\boldsymbol{z}_i) \cdot \boldsymbol{\chi} \quad . \tag{3.14}$$

Based on this approximation, mean and covariance are computed by

$$\mu_{x_i} = E[\hat{H}_x(\boldsymbol{z})] = E[H_x(\boldsymbol{z})] \tag{3.15}$$

$$\mathrm{cov}[x_i, x_j] = \mathrm{cov}[\hat{H}_x(\boldsymbol{z}_i), \hat{H}_x(\boldsymbol{z}_j)] = \mathrm{cov}[H_x(\boldsymbol{z}_i), \boldsymbol{\chi}] \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \mathrm{cov}^T[H_x(\boldsymbol{z}_j), \boldsymbol{\chi}] \quad . \tag{3.16}$$

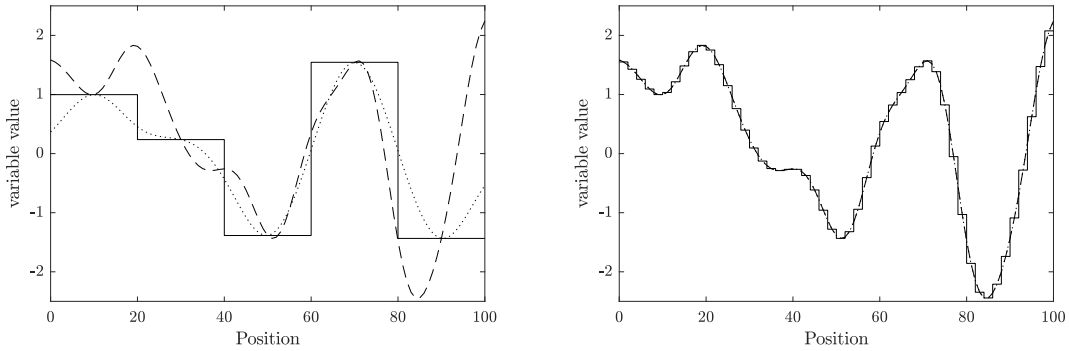


Figure 3.2: Different discretization methods for the same 1-D random field with the correlation length 10. The left figure shows the discretization on a coarse mesh, the right figure on a fine mesh. Legend: midpoint method (solid line), integration point method (dashed line), optimal linear estimator (dotted line) with control points at element centers.

Collecting all values in the covariance matrix leads to

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}} \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}}^{T} \tag{3.17}$$

with the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}}$ between discretized random variables $\boldsymbol{x}$ and the random field at the control points $\boldsymbol{\chi}$. Using the OLE formulation, only the matrices $\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{\chi}}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \in \mathbb{R}^{n_{\boldsymbol{\chi}} \times n_{\boldsymbol{\chi}}}$ must be stored instead of the full covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$. If the number of control points $n_{\boldsymbol{\chi}}$ is much smaller than the number of random variables $n_{\boldsymbol{x}}$, the memory requirements are significantly reduced. However, this comes at the cost of reduced accuracy of the random field representation.

As an alternative, approaches based on expansions, such as the Karhunen-Loève expansion [139] and the expansion optimal linear estimator method [138] are often applied to compute a memory efficient discretization of a random field. Using the continuous Karhunen-Loéve expansion, the spectral decomposition of the covariance function is computed, such that

$$\int_{\Omega} \boldsymbol{\Sigma}[H(\boldsymbol{z}_1), H(\boldsymbol{z}_2)] \phi_i(\boldsymbol{z}_1) d\boldsymbol{z}_2 = \lambda_i \varphi_i(\boldsymbol{z}_2) \tag{3.18}$$

with the i-th eigenfunction $\varphi_i(\boldsymbol{z}_2)$ and the corresponding eigenvalue λ_i. One can show that, the required number of eigenfunctions for reaching a given accuracy is bounded and depends on the correlation length. Especially for high correlation lengths, only very few eigenfunctions are required. However, computing the continuous Karhunen-Loéve expansion is only possible for special correlation functions and is hard to implement. Therefore, the discrete Karhunen-Loéve expansion is used more commonly in engineering. There, the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is decomposed in eigenvalues λ_i and eigenvectors $\boldsymbol{\varphi}_i$, such that

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i \quad . \tag{3.19}$$

Using the truncated Karhunen-Loéve expansion, only the M eigenvalues above a given threshold are considered. The eigenvectors are collected in the matrix $\boldsymbol{V}_{\boldsymbol{x}} \in \mathbb{R}^{n_{\boldsymbol{x}} \times M}$, the eigenvalues are

collected in the diagonal matrix $\boldsymbol{W}_{\boldsymbol{x}} \in \mathbb{R}^{M \times M}$ such that

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{V}_{\boldsymbol{x}} \boldsymbol{W}_{\boldsymbol{x}} \boldsymbol{V}_{\boldsymbol{x}}^T \quad . \tag{3.20}$$

Especially for very fine discretizations (e.g. in topology optimization), the required number of eigenvectors is very low (e.g. $M < \frac{n_{\boldsymbol{x}}}{100}$) leading to high memory savings. However, the original memory consuming covariance matrix still must be computed and the eigenvalue problem has to be solved for a large problem leading to a high demand on computational resources.

As an alternative, the expansion optimal linear estimator method (EOLE) proposed in [138] combines the advantages of the optimal linear estimator method and the Karhunen-Loéve expansion. Therefore, the OLE random field representation (3.14) with (3.12) and (3.13) is considered and the corresponding matrix $\boldsymbol{\Sigma}_{\boldsymbol{\chi}}$ is expanded by

$$\boldsymbol{\Sigma}_{\boldsymbol{\chi}} = \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}} \boldsymbol{V}_{\boldsymbol{\chi}}^T \quad . \tag{3.21}$$

Similar to (3.20), $\boldsymbol{V}_{\boldsymbol{\chi}} \in \mathbb{R}^{n_{\boldsymbol{\chi}} \times M}$ contains the eigenvectors and $\boldsymbol{W}_{\boldsymbol{\chi}} \in \mathbb{R}^{M \times m}$ the most relevant M eigenvalues of the matrix $\boldsymbol{\Sigma}_{\boldsymbol{\chi}}$. By application to (3.14), the optimal linear estimator approximation of the random field reads

$$\hat{H}_x(\boldsymbol{z}) = E[H_x(\boldsymbol{z})] - \operatorname{cov}[H_x(\boldsymbol{z}), \boldsymbol{\chi}] \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{-1} \boldsymbol{V}_{\boldsymbol{\chi}}^T \mu_{\boldsymbol{\chi}} + \operatorname{cov}[H_x(\boldsymbol{z}), \boldsymbol{\chi}] \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{-1} \boldsymbol{V}_{\boldsymbol{\chi}}^T \boldsymbol{\chi} \tag{3.22}$$

Using the definition of uncorrelated spectral random variables $\boldsymbol{\zeta}$ as

$$\boldsymbol{\chi} = \boldsymbol{E}[\boldsymbol{\chi}] + \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{0.5} \boldsymbol{\zeta} \tag{3.23}$$

the approximated random field simplifies to

$$\hat{H}(\boldsymbol{x}_x) = E[H_x(\boldsymbol{z})] + \boldsymbol{\Sigma}[H_x(\boldsymbol{z}), \boldsymbol{\chi}] \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{-0.5} \boldsymbol{\zeta} \quad , \tag{3.24}$$

and the discrete covariance matrix reads

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}} \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{-0.5} \boldsymbol{W}_{\boldsymbol{\chi}}^{-0.5} \boldsymbol{V}_{\boldsymbol{\chi}}^T \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}}^T \quad . \tag{3.25}$$

In consequence, only the matrix $\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{\chi}} \boldsymbol{V}_{\boldsymbol{\chi}} \boldsymbol{W}_{\boldsymbol{\chi}}^{-0.5} \in \mathbb{R}^{n_{\boldsymbol{x}} \times M}$ has to be stored. Besides, the original covariance matrix must not be computed and the corresponding eigenvalue problem is much smaller than for the Karhunen-Loéve expansion, leading to significantly reduced computation time and memory demands.

3.3 Conditioned random fields

In some situations, it is required to modify a random field such that boundary conditions are fulfilled. For example, it might be required that geometry imperfections are zero at the supports, since otherwise, the part can not be mounted. Therefore, the covariance matrix can be conditioned as described in [140, 141], where the variance is zero at the supports and grows to its prescribed value considering the correlation length. Besides, it is also possible to apply boundaries on the (spatial) gradient of the random variables (see e.g. [142]).

The basic concept is based on the optimal linear estimator method described before. In the first step, the OLE discretization of the random field considering only the clamped positions as control points $H_{x,con}(\boldsymbol{z}_i)$ with the covariance function

$$\operatorname{cov}[H_{x,con}(\boldsymbol{z}_i), H_{x,con}(\boldsymbol{z}_i)] = \operatorname{cov}[H_x(\boldsymbol{z}_i), \boldsymbol{\chi}] \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \operatorname{cov}[\boldsymbol{\chi}, H_x(\boldsymbol{z}_j)] \tag{3.26}$$

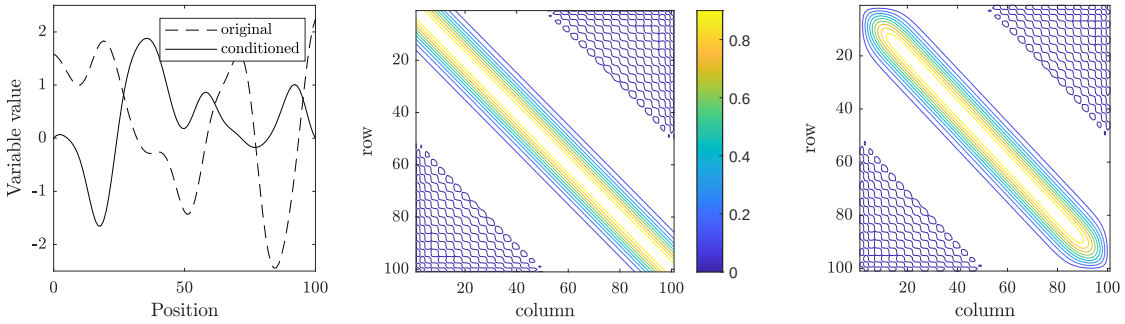


Figure 3.3: Left: Original and conditioned random field realizations of a 1D random field. Middle: Covariance matrix of the corresponding original random field. Right: Covariance matrix of the corresponding conditioned field.

is generated. Here, $\boldsymbol{\chi} = [H_x(\boldsymbol{z}_{con,1}), H_x(\boldsymbol{z}_{con,2}), \dots]^T$ represents the random field values evaluated at the control points $\boldsymbol{z}_{con,i}$, which are the positions of the boundary conditions. Per definition it holds

$$H_{x,con}(\boldsymbol{z}_{con,i}) = H_x(\boldsymbol{z}_{con,i}) \forall i \tag{3.27}$$

and hence, substracting $H_{x,con}$ from the original random field $H_x(\boldsymbol{z})$ leads to

$$H_x(\boldsymbol{z}) - H_{x,con}(\boldsymbol{z}) = 0 \quad \forall \boldsymbol{z} \in \{\boldsymbol{z}_{con,1}, \boldsymbol{z}_{con,2}, \dots\} \quad , \tag{3.28}$$

which fulfills the boundary conditions. Therefore, the conditioned random field $\hat{H}_x(\boldsymbol{z})$ is computed by

$$\hat{H}_x(\boldsymbol{z}) = H_x(\boldsymbol{z}) - H_{x,con}(\boldsymbol{z}) \tag{3.29}$$

with the covariance function

$$\begin{aligned} \mathrm{cov}[\hat{H}_x(\boldsymbol{z}_i), \hat{H}_x(\boldsymbol{z}_i)] &= \mathrm{cov}[H_x(\boldsymbol{z}_i), H_x(\boldsymbol{z}_i)] - \mathrm{cov}[H_{x,con}(\boldsymbol{z}_i), H_{x,con}(\boldsymbol{z}_i)] \\ &= \mathrm{cov}[H_x(\boldsymbol{z}_i), H_x(\boldsymbol{z}_i)] - \mathrm{cov}[H_x(\boldsymbol{z}_i), \boldsymbol{\chi}] \boldsymbol{\Sigma}_{\boldsymbol{\chi}}^{-1} \mathrm{cov}[\boldsymbol{\chi}, H_x(\boldsymbol{z}_j)] \end{aligned} \quad . \tag{3.30}$$

Following the framework, random fields and covariance functions as shown in figure 3.3 are generated. Whereas the original random field has arbitrary values at the positions 0 and 100, the conditioned field smoothly reaches 0 at these points. The corresponding covariance matrix is similar to the original one, but the variance reduces to zero for the first and the last variables.

A discretization of the random field and the corresponding discrete covariance matrix are found using the previously discussed discretization approaches, for example the midpoint method or the expansion optimal linear estimator method.

3.4 Probabilistic methods

Besides the basics on stochastics and the spatial representation of random variables, the current thesis also builds on probabilistic methods. Probabilistic methods are used to determine the stochastic properties (especially mean value μ_f and variance σ_f^2) of a function value $f(\boldsymbol{x})$ based on the

known stochastic properties of the uncertain input variables $\boldsymbol{x}$. In the current thesis, the Monte Carlo method and several methods based on Taylor series (i.e. first-order second-moment method, second-order fourth-moment method) are used and hence, derived and described in the following.

3.4.1 Monte Carlo method

The Monte Carlo method presented in [87, 143] is a method that has been widely used for solving complicated integrals, but also has been applied as a probabilistic method. The basic concept for using the Monte Carlo method as a probabilistic method is the following. From the known stochastic characteristics (e.g., stochastic moments, probability distribution, ...) of the input variables $\boldsymbol{x}$, many sets of variables $\boldsymbol{x}_1, \boldsymbol{x}_2, ...$ (called samples) are generated using a (pseudo-) random number generator. For every set of random variables, the objective function is evaluated ($f_1 = f(\boldsymbol{x}_1), f_2 = f(\boldsymbol{x}_2), ...$) and the resulting values f_1, f_2 are treated using empirical statistical methods. For example, mean μ_f and standard deviation σ_f of the response f are computed using (3.3) and (3.4).

Since most random number generators are not able to consider correlations between different variables, the generated uncorrelated variables must be transformed to correlated variables. Using the expansion optimal linear estimator method, the variables $\boldsymbol{\zeta}$ in (3.24) can directly be generated using the uncorrelated random number generator, since these variables are already uncorrelated. Otherwise, the random variables are generated by [143]

$$\boldsymbol{x} = \boldsymbol{\Sigma}^{\frac{1}{2}} \boldsymbol{\zeta} = \boldsymbol{V}_{\boldsymbol{x}} \boldsymbol{W}_{\boldsymbol{x}}^{\frac{1}{2}} \boldsymbol{\zeta} \tag{3.31}$$

with the uncorrelated variables $\boldsymbol{\zeta}$, the matrix of eigenvectors $\boldsymbol{V}_{\boldsymbol{x}}$ and the matrix of eigenvalues $\boldsymbol{W}_{\boldsymbol{x}}$.

3.4.2 Taylor series-based methods

The Taylor series-based methods, such as the first-order second-moment method and the second-order-fourth moment method derived in [144] are based on the Taylor approximation

$$T_f(\boldsymbol{x}) = f(\mu_{\boldsymbol{x}}) + \sum_i \frac{\partial f}{\partial x_i}(x_i - \mu_{x_i}) + \frac{1}{2}\sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j}(x_i - \mu_{x_i})(x_j - \mu_{x_j}) + \dots \tag{3.32}$$

of the original objective function f. Inserting (3.32) into the analytic equation of the mean (3.1) leads to

$$\begin{aligned} \mu_f = f(\mu_{\boldsymbol{x}}) \int_{-\infty}^{\infty} p(\boldsymbol{x}) d\boldsymbol{x} + \sum_i \frac{\partial f}{\partial x_i} \int_{-\infty}^{\infty} (x_i - \mu_{x_i}) p(x_i) dx_i \\ + \frac{1}{2}\sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x_i - \mu_{x_i})(x_j - \mu_{x_j}) p(x_i, x_j) dx_i dx_j + \dots \end{aligned} \quad . \tag{3.33}$$

Since $\int_{-\infty}^{\infty} p(\boldsymbol{x}) d\boldsymbol{x} = 1$, $\int_{-\infty}^{\infty} (x_i - \mu_{x_i}) p(x_i) dx_i = 0$ and $\int_{-\infty}^{\infty} (x_i - \mu_{x_i})(x_j - \mu_{x_j}) p(x_i, x_j) dx_i dx_j = \mathrm{cov}(x_i, x_j)$, the whole equation simplifies to

$$\mu_f = f(\mu_{\boldsymbol{x}}) + \frac{1}{2}\sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j} \mathrm{cov}(x_i, x_j) + \dots \quad . \tag{3.34}$$

The variance may be derived similarly, using (3.2).

Chapter 4

Scalable approaches for optimization under uncertainty

Combining the basics of structural optimization with the basics of stochastic and probabilistic leads to robust design optimization techniques. In the following, new scalable formulations for robust design optimization using the first-order second-moment method (publication 1) and the second-order fourth-moment method (publication 2) are developed and tested at different benchmark examples. There, a major focus is set on the computation of the gradients of the approximated stochastic moments, since this part usually is computationally costly. Afterwards, a new approach tailored to problems which are prone to buckling is presented and compared to the classic first-order second-moment method considering geometric nonlinear problems.

4.1 An efficient and non-intrusive approach for robust design optimization with the first-order second-moment method

The first publication [145] contributes to the first-order second-moment method, which can be derived as outlined in 3.4.2 using a first-order Taylor series. As a probabilistic approach, the first-order second-moment method is known to be rather inaccurate [114, 146, 147], if the objective function behaves truly nonlinear near to the mean state. However, there are highly efficient formulations in the context of robust design optimization [115], which do not increase the computational cost compared to a deterministic optimization. Therefore, the required resources are much lower compared (usually factor 100 to infinity) to more accurate approaches. Besides, the lack of accuracy is not critical in the context of a robust optimization if the final design is evaluated using a more accurate approach, proving or disproving the increased robustness.

One main challenge in the context of the first-order second-moment method is to compute the gradients of the stochastic moments efficiently for arbitrary problems. Therefore, the following publication compares different existing formulations and proposes a new non-intrusive approach called "principal sensitivity first-order second-moment method" which shows high computational and memory efficiency, but can also be applied to arbitrary problems as a black-box method.

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 414 (2023) 116136
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An efficient and non-intrusive approach for robust design optimization with the first-order second-moment method

Jan Christoph Krüger[a,*], Micah Kranz[a], Timo Schmidt[b], Robert Seifried[b], Benedikt Kriegesmann[a]

[a] *Institute of Structural Mechanics for Lightweight Design, Hamburg University of Technology, Hamburg, Germany*
[b] *Institute of Mechanics and Ocean Engineering, Hamburg University of Technology, Hamburg, Germany*

Received 9 February 2023; received in revised form 26 April 2023; accepted 19 May 2023
Available online 8 June 2023

Abstract

A modified robust design optimization approach is presented, which uses the first-order second-moment method to compute the mean value and the standard deviation for arbitrary objective functions. Existing approaches compute the gradient of the variance using the adjoint method, direct differentiation or finite differences, respectively. These approaches either access to the FE-code and/or have high computational cost. In this paper, a new approach for the computation of the gradient of the variance is provided. It can be easily implemented as a non-intrusive method, which behaves similar to finite differences with the cost of only one additional objective evaluation, independent of the number of variables. Here, a step-size has to be chosen carefully and therefore, a procedure to determine a problem-independent step-size is provided. As an alternative, the approach can be implemented as an analytic method with the same cost like the adjoint method, but providing wider applicability (e.g. eigenvalue problems). The provided approach is derived, analyzed and applied to several benchmark examples.

Keywords: Robust design optimization (RDO); Taylor series expansion; Robust topology optimization; Method of moments

1. Introduction

The presence of uncertainties (like scattering material properties and loads) and the fact that considering these uncertainties in design optimization influences the optimization result have led to a large amount of publications during the last decades. Overviews are for instance given by Park et al. [1], Schuëller and Jensen [2] and more recently by Kanno [3]. While there are multiple approaches to tackle uncertainties, the current paper focuses on probabilistic approaches. In low dimensional problems, surrogate models may be trained to perform the optimization and/or the uncertainty quantification [4]. However, high dimensional problems like topology and non-parametric shape optimization require the use of gradient-based optimization approaches [5].

Among the first works where uncertainty was introduced into topology optimization were [6–8]. All these early works considered reliability-based optimization with random loads. Random loads typically represent a small number of random parameters, and when considering linear problems, Monte Carlo approaches can be efficiently

* Corresponding author.
E-mail address: jan.krueger@tuhh.de (J.C. Krüger).

https://doi.org/10.1016/j.cma.2023.116136
0045-7825/

used by superposing randomly scaled unit load cases. This is not possible for other random effects like spatially varying material properties and geometric deviations, as considered in [9,10]. These works solved robust design optimization problems, i.e. minimized a combination of mean and standard deviation of the objective function. Using the Monte Carlo method to solve this problem as in [9] increases the computational cost by the number of realizations used, typically hundreds or thousands. As an alternative, Keshavarzzadeh proposes the use of the non-intrusive polynomial chaos expansion in [11]. Using this approach, the computational cost and its accuracy depends on the order of the considered polynomial and the number of random variables. However, even for a first-order polynomial, one coefficient per random variable has to be computed. This leads to at least one function evaluation per random variable. Using higher order polynomials, the computational cost increases superlinear with respect to polynomial order and the number of random variables. Hence, more efficient approaches based on Taylor series expansions have been suggested. The perturbation approach is based on a Taylor series expansion of the equilibrium conditions. Doltsinis et al. [12,13] embedded the perturbation approach into robust design optimization framework, and later Lazarov et al. [10] applied it to topology optimization. Here, the number of systems of equations to be solved and hence, the computational cost, scales with the number of random parameters. In particular, it scales linearly for a first-order expansion and quadratically in case of a second-order expansion. Hence, Lazarov et al. [10] used the discrete Karhunen–Loève transformation to reduce the number of parameters describing the discretized random fields.

In Kriegesmann and Lüdeker [14], the use of the first-order second-moment (FOSM) method for robust topology optimization is proposed. In difference to the perturbation approach, the probabilistic objective function is expanded. When considering compliance as objective function, this approach requires the solution of only one additional adjoint system per iteration, independently of the number of random parameters. When using a second-order expansion, the computational cost is proportional to the number of random parameters. However, for random geometries and material properties the first order expansion turned out to provide sufficiently good results to achieve a robust design. Kranz et al. [15] provided a generalized formulation of the approach, which also allows to consider other objective functions like a certain displacement or the maximum stress. Then, the FOSM-based approach requires the solution of three additional adjoint systems compared to a deterministic analysis. Though this is a manageable number of equations to be solved, the computational cost of this approach is driven by computing large non-sparse matrices, which turns out to be significant for stress-based problems. Another challenge occurs when considering eigenvalue problems. Applying the same approach from [14] to the eigenfrequency of a structure provides an adjoint system that cannot be solved [16].

All aforementioned approaches are highly intrusive and hence, not available in commercial code. In difference to that, many combinations of objective functions (compliance, max. stress, eigenfrequency, ...) and design parameters (sizing, shape and topology optimization) are possible in commercial codes, i.e. the gradients with respect to the design variables are provided. This motivated the work of Steltner et al. [17], who suggested a semi-intrusive approach which makes use of the deterministic design gradients and approximates the derivatives as well as the mixed partial second-order derivatives required for FOSM by finite differences. The approach is implemented as a plug-on for commercial codes and can easily be applied to any objective function and even nonlinear problems [18]. The computation cost however scales linearly with the number of random parameters. Therefore, when considering spatial variations such as geometric imperfections, again, reduction techniques like discrete Karhunen–Loève transformation need to be applied [18].

The current paper proposes a new approach for robust design optimization based on the first-order second-moment method called *principal sensitivity first-order second-moment method*. It delivers an efficient framework which combines the advantages of the aforementioned approaches. It is derived as an intrusive and an non-intrusive version. These approaches require less computational time and/or less memory than all existing approaches based on the first-order second moment method. Furthermore, eigenvalue problems, which could not be solved before, can be tackled with these new methods.

The paper is organized as follows: In Section 2, the general first-order second-moment framework is described and existing formulations are discussed. In Section 3, the new approach, called principal sensitivity first-order second-moment method, is derived and different aspects are analyzed. In Section 4, the new framework is compared to existing frameworks for different numerical examples including topology optimization considering stress, eigenfrequencies and contact.

2. Robust design optimization with the first-order second-moment method

Robust design optimization is similar to standard design optimization. The same parametrization, optimization algorithms, FE-solvers and problem dependent aspects (e.g. penalization in topology optimization) are considered. However, in a robust design optimization the objective function contains the mean μ_f and standard deviation σ_f of some deterministic objective f. These values are computed using probabilistic methods. The robust objective is therefore defined as

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

with the random parameters $\boldsymbol{x}$ and the design parameters $\boldsymbol{y}$. The weighting factor κ can be chosen arbitrarily. However, the choice of κ significantly influences the results of a design optimization. In general, a large weighting factor forces the optimization to a design with a small standard deviation at the cost of a worse mean. In difference, a small weighting factor focuses the optimization to the mean value. The first-order second moment method (FOSM) was proposed by Cornell in [19] and is based on a Taylor-series expansion of the objective function. Using this approach, mean and variance are approximated using

$$\mu_f = f(\mu_{\boldsymbol{x}}) \tag{2}$$

$$\sigma^2{}_f = \frac{df}{d\boldsymbol{x}}^T(\mu_{\boldsymbol{x}}) \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}(\mu_{\boldsymbol{x}}) \tag{3}$$

with the mean vector $\mu_{\boldsymbol{x}}$ of the random parameters $\boldsymbol{x}$ and the covariance matrix $\boldsymbol{Cov}$ of the random parameters [14]. It is assumed that the covariance matrix is independent of design variables. (A design-dependent covariance requires the consideration of additional terms in the gradient derivation, which is extensively discussed in [20]. These terms do not influence the subsequent consideration and therefore, this discussion is not repeated in this paper.) In order to use gradient based optimization algorithms the derivatives of mean and variance are computed by

$$\frac{d\mu_f}{d\boldsymbol{y}} = \frac{df}{d\boldsymbol{y}}(\mu_{\boldsymbol{x}}) \tag{4}$$

and

$$\frac{d\sigma^2{}_f}{d\boldsymbol{y}} = 2\frac{d^2 f}{d\boldsymbol{y} d\boldsymbol{x}} \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}} \quad . \tag{5}$$

The gradient of the standard deviation is given by

$$\frac{d\sigma_f}{d\boldsymbol{y}} = \frac{d\sigma_f^2}{d\boldsymbol{y}} \cdot \frac{1}{2 \cdot \sigma_f} \quad . \tag{6}$$

Mean, standard deviation and gradient of the mean can be computed straight forward since first-order sensitivities of the objective are already required for a deterministic optimization. Computing the gradient of the variance is however challenging since second-order sensitivities are required.

2.1. Existing methods for determining the gradient of the variance

The current section summarizes existing methods to compute the gradient of the variance. The approaches either calculate the mixed partial derivative $\frac{d^2 f}{d\boldsymbol{y} d\boldsymbol{x}}$ or the variance gradient directly. Numeric methods as well as analytic approaches can be used.

2.1.1. Finite differences

Considering commercial code with integrated optimization methods, only the function value and gradients with respect to design variables are available. In order to compute the variance and variance sensitivities, first- and second-order derivatives with respect to random variables have to be computed using finite differences like in [18]. The required approximations of the sensitivities are given by

$$\frac{df}{dx_i} = \frac{f(x_i + \Delta) - f(x_i)}{\Delta} \tag{7}$$

$$\frac{d^2 f}{dx_i d\boldsymbol{y}} = \frac{\frac{df}{d\boldsymbol{y}}(x_i + \Delta) - \frac{df}{d\boldsymbol{y}}(x_i)}{\Delta} \quad . \tag{8}$$

with the random parameter x_i and a finite step Δ chosen suitable small.

However, if it is possible to extract the gradient with respect to random parameters from the FE-code, Eq. (7) is not required. As an alternative, the gradient of the standard deviation might be computed using finite difference directly to the standard deviation by

$$\frac{d\sigma_f}{dy_i} = \frac{\sigma_f(y_i + \Delta) - \sigma_f(y_i)}{\Delta} \quad . \tag{9}$$

Finite difference interprets the FE-model as a black-box problem. Therefore, the formulas are problem independent and non-intrusive. However, this method has some drawbacks. Finite-differences are an approximation of the true sensitivities. This leads to numerical errors, which can be reduced using small step-sizes. However, very small step-sizes lead to numerical errors due to a limited machine accuracy. In order to compute a whole gradient, finite differences require an objective evaluation for every considered parameter. Especially for problems with many parameters (e.g. topology optimization) this ends up in a extremely high computation time. Therefore, this method cannot be used in realistic examples.

2.1.2. Direct differentiation

In difference to finite-differences, direct differentiation is used to compute the second order derivative $\frac{d^2 f}{dx_i d\boldsymbol{y}}$ analytically. The general objective function is given by

$$f = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}_0) \quad , \tag{10}$$

where the implicit parameter $\boldsymbol{u}_0$ is computed as the solution of a (possibly nonlinear) system of equations

$$\boldsymbol{0} = \boldsymbol{R}_0(\boldsymbol{u}_0, \boldsymbol{x}, \boldsymbol{y}) \quad . \tag{11}$$

A simple example would be the standard linear finite element problem, where $\boldsymbol{R}$ is given by

$$\boldsymbol{R}_0 = \boldsymbol{K}\boldsymbol{u}_0 - \boldsymbol{f} \quad , \tag{12}$$

with the stiffness matrix $\boldsymbol{K}$, the deformation vector $\boldsymbol{u}_0$ and the external force vector $\boldsymbol{f}$. It is assumed that the first-order objective gradient with respect to design parameters $\boldsymbol{y}$ is already given for a deterministic optimization. Often, the objective gradient depends on implicit adjoint variables $\boldsymbol{\lambda}_0$. Therefore, one obtains the general system of equations

$$\frac{df}{d\boldsymbol{y}} = \frac{df}{d\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \tag{13}$$

$$\boldsymbol{0} = \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}) \tag{14}$$

with the new implicit (nonlinear) system of equations $\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y})$ and the explicit objective gradient $\frac{df}{d\boldsymbol{y}}$. The implicit variable $\boldsymbol{u}$ contains the previously defined variables $\boldsymbol{u}_0$ as well as the adjoint variables $\boldsymbol{\lambda}_0$. Differentiation with respect to $\boldsymbol{x}$ of both equations leads to

$$\frac{d^2 f}{d\boldsymbol{y} d\boldsymbol{x}} = \frac{d^2 f}{d\boldsymbol{y} d\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}, \frac{d\boldsymbol{u}}{d\boldsymbol{x}}) \tag{15}$$

$$\boldsymbol{0} = \frac{d\boldsymbol{R}}{d\boldsymbol{x}}(\boldsymbol{u}, \boldsymbol{x}, \frac{d\boldsymbol{u}}{d\boldsymbol{x}}) \quad . \tag{16}$$

First, the implicit equations (16) are solved to obtain the gradient of the state-variables $\boldsymbol{u}$. Afterward, the gained state-variable gradient is inserted into (15). The computed second-order derivative of the objective function is inserted into (5) to obtain the variance gradient. This method delivers an exact value of the gradient of the variance. However, there are some drawbacks. Solving the implicit equations (16) requires the solution of one (nonlinear) system of equations per design variable, since the derivative of every entry of the vector $\boldsymbol{u}$ has to be found with respect to every random variable x_i. For example, in the compliance problem

$$\boldsymbol{0} = \frac{d\boldsymbol{R}}{dx_i} = \frac{d\boldsymbol{K}}{dx_i}\boldsymbol{u} - \boldsymbol{K}\frac{d\boldsymbol{u}}{dx_i} \tag{17}$$

has to be solved for every index i. Especially for problems with many parameters, this results in a very high computation time and storage demand. The computational cost can be reduced using some matrix decomposition of the stiffness matrix, since only the right-hand side of the system of equation changes for different indices i. This modification however increases the storage demand further. The computational cost is still much higher than a deterministic function evaluation. Another drawback is that the derived equations in detail look differently for each combination of objective function, random parameters and design parameters. Hence, the implementation effort is very high.

2.1.3. Adjoint method

Using the discrete adjoint method by [21], the conditions are similar to the direct differentiation, considering the objective function (10) (11). The objective gradient with respect to random variables $\boldsymbol{x}$ is again obtained by (13) (14) using the system of equations

$$\frac{df}{d\boldsymbol{x}} = \frac{df}{d\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}) \tag{18}$$

$$\boldsymbol{0} = \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}) \tag{19}$$

with the implicit system of equations $\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y})$ and the explicit objective gradient $\frac{df}{d\boldsymbol{x}}$. The implicit system of equations again contains the residual equations of a FE-problem as well as previously used adjoint systems. Different to direct differentiation, this system of equations is not differentiated. Instead, the variance equation (3) is expanded by the implicit equation (19) to

$$\sigma^2{}_f = \frac{df}{d\boldsymbol{x}}^T \boldsymbol{Cov} \frac{df}{d\boldsymbol{x}} + \boldsymbol{\lambda}^T \cdot \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}) \tag{20}$$

with the adjoint vector $\boldsymbol{\lambda}$. Differentiation leads to

$$\frac{d\sigma^2}{d\boldsymbol{y}} = 2 \cdot \frac{df}{d\boldsymbol{x}}^T \boldsymbol{Cov} \left(\frac{d\partial f}{d\boldsymbol{x}\partial \boldsymbol{y}} + \frac{d^2 f}{d\boldsymbol{x} d\boldsymbol{u}} \frac{d\boldsymbol{u}}{d\boldsymbol{y}} \right) + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} + \boldsymbol{\lambda}^T \frac{d\boldsymbol{R}}{d\boldsymbol{u}} \frac{d\boldsymbol{u}}{d\boldsymbol{y}} \quad . \tag{21}$$

With some reordering one gets

$$\frac{d\sigma^2}{d\boldsymbol{y}} = 2 \cdot \frac{df}{d\boldsymbol{x}}^T \boldsymbol{Cov} \frac{d\partial f}{d\boldsymbol{x}\partial \boldsymbol{y}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} \tag{22}$$

with the adjoint system

$$\frac{d\boldsymbol{R}}{d\boldsymbol{u}}^T \boldsymbol{\lambda} + 2 \cdot \frac{d^2 f}{d\boldsymbol{x} d\boldsymbol{u}}^T \boldsymbol{Cov} \frac{df}{d\boldsymbol{x}} = \boldsymbol{0} \quad . \tag{23}$$

The adjoint method is problem-dependent. For a linear compliance example, the objective gradient is given by $\frac{df}{d\boldsymbol{x}} = -\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \boldsymbol{u}$ and the residual equation by $\boldsymbol{R} = (\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f})$. Using Eq. (22) leads to

$$\frac{d\sigma^2}{d\boldsymbol{y}} = 2 \cdot \left(\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \boldsymbol{u} \right)^T \boldsymbol{Cov} \left(\boldsymbol{u}^T \frac{d^2 \boldsymbol{K}}{d\boldsymbol{y} d\boldsymbol{x}} \boldsymbol{u} \right) + \boldsymbol{\lambda}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \boldsymbol{u} \tag{24}$$

with the adjoint system according to (23)

$$\boldsymbol{K}\boldsymbol{\lambda} + 4 \cdot \left(\frac{d\boldsymbol{K}}{d\boldsymbol{x}} \boldsymbol{u} \right)^T \boldsymbol{Cov} \left(\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \boldsymbol{u} \right) = \boldsymbol{0} \quad . \tag{25}$$

A big advantage of the adjoint method compared to finite differences and direct differentiation is, that it produces exact gradient values at the cost of only one additional system of equations, independent of the number of variables. Therefore, the adjoint method is very efficient. On the other hand, the adjoint method is problem dependent like direct differentiation. Therefore, it suffers the same high implementation effort like direct differentiation. Although the method is computationally efficient, the required partial derivatives for certain objective function require a large amount of storage and computation time. Especially for stress problems, this effect outweighs the efficient formulation [15].

3. Robust design optimization with the principal sensitivity first-order second-moment method

All previously shown methods have some significant drawbacks like extremely high computation time, storage demand and an intrusive nature. Therefore the authors propose a new method called principal sensitivity first-order second-moment method (PSF), which overcomes the drawbacks of existing methods for calculating the gradient of the variance estimated by the first-order second-moment method. The basic idea is neither to compute the second-order derivative $\frac{d^2 f}{d\boldsymbol{y}d\boldsymbol{x}}$, nor to expand the variance formula by an adjoint problem. Instead, the product $\frac{d^2 f}{d\boldsymbol{y}d\boldsymbol{x}}\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}$ is computed directly. We present two versions of the PSF approach, namely the non-intrusive, numeric principal sensitivity FOSM (nPSF) and an intrusive analytic principal sensitivity FOSM (aPSF).

3.1. Numeric principal-sensitivity FOSM

The numeric principal sensitivity FOSM approach is based on a Taylor series of the objective gradient with respect to design variables $\boldsymbol{y}$ given by

$$\frac{df}{d\boldsymbol{y}}(\mu_x+\Delta\boldsymbol{x})=\frac{df}{d\boldsymbol{y}}(\mu_x)+\frac{d^2 f}{d\boldsymbol{y}d\boldsymbol{x}}(\mu_x)\cdot\Delta\boldsymbol{x}+\mathcal{O}(\Delta\boldsymbol{x}^2) \tag{26}$$

with a step $\Delta\boldsymbol{x}$ and terms of at least order two $\mathcal{O}(\Delta\boldsymbol{x}^2)$. Choosing $\Delta\boldsymbol{x}$ to be

$$\Delta\boldsymbol{x}=\epsilon\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x) \quad . \tag{27}$$

with an arbitrary scaling factor ϵ leads to the Taylor series

$$\frac{df}{d\boldsymbol{y}}(\mu_x+\epsilon\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x))=\frac{df}{d\boldsymbol{y}}(\mu_x)+\frac{d^2 f}{d\boldsymbol{y}d\boldsymbol{x}}(\mu_x)\cdot\epsilon\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x)+\mathcal{O}(\Delta\boldsymbol{x}^2) \quad . \tag{28}$$

We call the vector $\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x)$ the *principal sensitivity* direction. The reason for this name and the properties of this vector are discussed in Section 3.1.1. Reordering Eq. (28) yields

$$\begin{aligned}&\frac{d^2 f}{d\boldsymbol{y}d\boldsymbol{x}}(\mu_x)\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x)\\&=\frac{1}{\epsilon}\left[\frac{df}{d\boldsymbol{y}}\left(\mu_x+\epsilon\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x)\right)-\frac{df}{d\boldsymbol{y}}(\mu_x)\right]+\mathcal{O}(\left[\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}\right]^2)\frac{\mathcal{O}(\epsilon^2)}{\epsilon} \quad .\end{aligned} \tag{29}$$

Note that (29) is almost identical to the gradient of the variance given by (5) (besides a factor of two and terms of order $\mathcal{O}(\left[\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}\right]^2)\mathcal{O}(\epsilon)$). Neglecting higher order terms, the gradient of the variance is approximated by

$$\frac{d\sigma^2}{d\boldsymbol{y}}\approx\frac{2}{\epsilon}\left[\frac{df}{d\boldsymbol{y}}\left(\boldsymbol{x}_{pert}\right)-\frac{df}{d\boldsymbol{y}}(\mu_x)\right] \quad . \tag{30}$$

with the perturbed variables $\boldsymbol{x}_{pert}$ given by

$$\boldsymbol{x}_{pert}=\mu_x+\epsilon\cdot\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}(\mu_x) \quad . \tag{31}$$

The error of this approximation is caused by higher order terms. Since $\boldsymbol{Cov}\cdot\frac{df}{d\boldsymbol{x}}$ is independent of ϵ, it is expected that the error scales linearly with the scaling factor ϵ for sufficiently small steps. Eq. (30) shows that the gradient of the variance can be estimated by evaluating the gradient of the deterministic objective f for two realizations of the random vector, one given by the mean vector and another one given by the principal sensitivity direction. This method is non-intrusive and easy to implement, as long as the gradients $\frac{df}{d\boldsymbol{y}}$ and $\frac{df}{d\boldsymbol{x}}$ are supplied by the used (e.g. commercial) software. In that case, numeric principal sensitivity FOSM can be implemented as a plugon, which evaluates the objective at the two desired points. Using nPSF, the computation time is only doubled compared to a deterministic optimization, while the storage demand does not increase significantly.

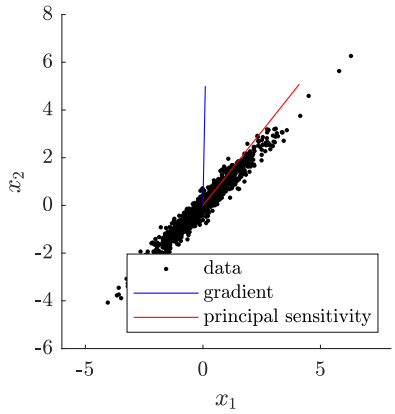


Fig. 1. Principal sensitivity and gradient for example function $0.1x_1+5x_2$ and 80% correlated standard uniform distributed random parameters.

3.1.1. Interpretation of principal sensitivity direction

The principal sensitivity direction $\boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}(\mu_x)$ is used for the gradient determination within the principal sensitivity FOSM. It combines the principal components of the covariance matrix with the local objective behavior. This ends up in a direction near the principal components, i.e. the eigenvectors of the covariance matrix, and with some shift towards the gradient. This is visualized in Fig. 1 for the example function $0.1x_1 + 5x_2$ and the covariance matrix $\boldsymbol{Cov} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$. Since FOSM assumes the objective function to be linear with respect to random variables, the variance given by (3) as $\sigma_f^2 = \frac{df}{d\boldsymbol{x}}^T \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}$ can be interpreted as a two point function evaluation. With the first-order Taylor series

$$F(\boldsymbol{y}, \Delta\boldsymbol{x}) = f(\boldsymbol{y}, \mu_x) + \frac{df}{d\boldsymbol{x}}(\boldsymbol{y}, \mu_x) \cdot \Delta\boldsymbol{x}(\boldsymbol{y}) \tag{32}$$

the variance can be written as

$$\sigma^2 = F(\boldsymbol{y}, \Delta\boldsymbol{x}) - F(\boldsymbol{y}, \boldsymbol{0}) \tag{33}$$

where $\Delta\boldsymbol{x}$ equals the principal sensitivity direction $\boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}$. This means that the principal sensitivity direction is something like the effective, one-dimensional scatter of a multivariate random parameter under consideration of the objective behavior. A linear approximation of the objective function is only exact for some combinations of objective function and random parameters. One example is the optimization for minimal tip displacement under a random load. For other cases the first-order approximation still provides satisfying results if the imperfections are small. In the case of moderate imperfections or highly non-linear functions, significant errors are caused when using a linear approximation. In these cases, higher-order methods or sampling based methods (e.g. Monte Carlo) are preferred [22].

For differentiation of σ^2 the derivative of the two point function evaluation (33) has to be considered. This leads to the formula of numeric principal sensitivity FOSM with a scaling factor $\epsilon = 1$.

3.1.2. Aspects of scaling factor ϵ

Without a scaling factor mismatching units are combined in the numeric principal sensitivity FOSM. If one considers the units of the objective $[f] = \mathcal{F}$ and the random parameters $[\boldsymbol{x}] = \mathcal{X}$, it follows that the unit of the Covariance is $[\boldsymbol{Cov}] = \mathcal{X}^2$ and the unit of the objective gradient is $[\frac{df}{d\boldsymbol{x}}] = \mathcal{F}/\mathcal{X}$. This means that the principal sensitivity direction $[\boldsymbol{Cov}\frac{df}{d\boldsymbol{x}}] = \mathcal{F}\mathcal{X}$ and the random parameters have different units. Since they are added in order to compute a disturbed evaluation point, the approximation in (30) is ill conditioned, though valid from a mathematical point of view. In order to determine accurate values for the gradient of the variance, the scaling factor should be chosen appropriately small. Since the norm of the principal sensitivity is design-dependent and therefore

the step size would change in every iteration, a normalized scaling factor $\hat{\epsilon}$ is defined by

$$\hat{\epsilon} = \epsilon \cdot \left\| \boldsymbol{Cov} \frac{df}{d\boldsymbol{x}} \right\| \quad . \tag{34}$$

For the use of nPSF, the normalized scaling factor $\hat{\epsilon}$ is fixed to a certain value. The scaling factor ϵ is computed using (34) in every iteration. Inserting (34) into (29) leads to an error estimator

$$\text{error} = \frac{\mathcal{O}(\left[\boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}\right]^2)}{\mathcal{O}(\left\|\boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}\right\|)} \mathcal{O}(\hat{\epsilon}) \quad , \tag{35}$$

which scales linearly with respect to the normalized scaling factor. Hence, PSF has similar behavior like finite differences. This assumption is supported by Fig. 2. The relative errors of the gradient of the variance, computed with nPSF and finite differences, are shown for the distributed load example of chapter 4.1. Different discretizations and normalized scaling parameters $\hat{\epsilon}$ are considered. The fine discretization consists of 22500 elements, the coarse discretization contains 225 finite elements. Double-precision numbers are used. This example is chosen to illustrate the following discussion, while the same observation is made for other examples. The relative error e_i of the ith entry of the gradient vector $\frac{d\sigma^2}{dy_i}$ is defined by

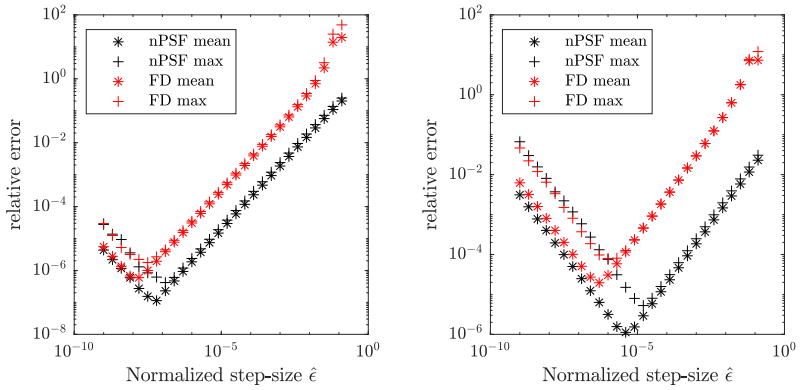
$$e_i = \frac{\left| \frac{d\sigma^2}{dy_i}_{PSF} - \frac{d\sigma^2}{dy_i}_{a} \right|}{\frac{d\sigma^2}{dy_i}_{a}} \tag{36}$$

with the estimator based on PSF $\frac{d\sigma^2}{dy_i}_{PSF}$ and the analytic value $\frac{d\sigma^2}{dy_i}_{a}$. The analytic gradient is computed using the adjoint method and verified using direct differentiation. In order to visualize the error of the gradient in one figure, mean value and maximum value of the element-wise errors are shown. The double-logarithmic plot in Fig. 2 shows, that the error decreases linearly up to a certain step-size. Afterwards, the error increases linearly. This observation confirms the assumption of a convergence order of one for PSF and finite differences. The increasing error for small step-sizes is caused by rounding errors. Comparing both plots reveals that the rounding errors become much more relevant for models with more elements using nPSF. The minimum mean error is obtained for the low discretization model at $\hat{\epsilon} = 6 \cdot 10^{-8}$ and for the high discretization model at $\hat{\epsilon} = 4 \cdot 10^{-6}$. The difference is about the same as the square root of the difference in numbers of finite elements. Using finite differences, this effect is not observed that clearly.

A reason for this behavior is illustrated using the 22 500 element example. Models with a high amount of elements in general have random variable vectors with more entries. Since the norm of the perturbation vector $\Delta\boldsymbol{x}$ is constrained to the value of $\hat{\epsilon}$ independently of the number of elements, every entry becomes smaller. If one assumes that every entry of the principal sensitivity is identical, all values of $\Delta\boldsymbol{x}$ would be $1/150 \approx 7 \cdot 10^{-3}$ to match $\hat{\epsilon} = 1$. At a normalized scaling factor of $\hat{\epsilon} = 10^{-6}$, every entry is reduced to values around $\Delta x_i = 7 \cdot 10^{-9}$. This perturbation vector $\Delta\boldsymbol{x}$ is added to the random variable vector $\boldsymbol{x}$ with entries around 1 for the considered example. Using double precision numbers, the machine accuracy is at about $\alpha_m = 10^{-16}$. This means that the numeric inaccuracy quotient is $\frac{\alpha_m}{\Delta x_i} = 10^{-7}$ which is of a similar order as the mean error at a normalized scaling factor.

At finite differences, only one entry of the random variable vector $\boldsymbol{x}$ is disturbed by the whole step size. Hence, the error of the random variable $\frac{\alpha_m}{\Delta x_i} = 1 \cdot 10^{-11}$ is not significant. However, the impact of rounding errors increases for bigger models. A reason is given by rounding errors of the objective value. If the number of elements increases, the influence of one single element decreases. Therefore, the change in the objective function becomes very small. Since finite differences use the difference in objective values, rounding errors become significant. Similar observations are done in [23].

For large-scale examples it is possible that there is no acceptable trade-off between rounding errors and truncation errors. Fig. 2 reveals that this holds for PSF as well as for finite differences. One solution is to use extended precision numbers, which is not common in commercial software or standard programming languages. Another approach is to derive a complex version of PSF, similar to complex finite differences given in [24]. However, the same aspects like described for extended precision numbers hold for complex differences.



(a) Error of example discretized with 225 elements.

(b) Error of example discretized with 22500 elements.

Fig. 2. Relative error of gradient estimation using nPSF and finite differences for distributed load topology optimization example in 4.1 with different discretization.

Alternatively, it is possible to derive a higher order approximation of the gradient of the variance. Similar to central differences, a PSF approximation with second-order convergence behavior is given by

$$\frac{d\sigma^2}{d\boldsymbol{y}} \approx \frac{1}{\epsilon}\left[\frac{df}{d\boldsymbol{y}}\left(\mu_x + \epsilon \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}(\mu_x)\right) - \frac{df}{d\boldsymbol{y}}\left(\mu_x - \epsilon \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}(\mu_x)\right)\right] \quad . \tag{37}$$

The second-order convergence behavior causes less truncation errors for large steps. Therefore, it is possible to tackle larger models using this approach. The computational cost increases to two instead of one objective evaluations, independent of the number of variables.

3.2. Analytic principal sensitivity FOSM

In the previous section, the numeric principal sensitivity FOSM is derived. It is very easy to implement and only doubles the computation time compared to a deterministic optimization. However, it still delivers an approximation of the gradient of the variance. Further investigations show that Eq. (30) is a finite difference step to compute the directional derivative of the objective function into the non-Cartesian direction $\boldsymbol{Cov}\frac{df}{d\boldsymbol{x}}$. The same observation can be done for the analytic formula (5), since direction derivatives are generally calculated for a differentiable function $\tilde{f}$ using

$$\frac{d\tilde{f}}{ds} = \frac{d\tilde{f}}{d\boldsymbol{x}} \cdot \boldsymbol{s} \tag{38}$$

when $\boldsymbol{s}$ is part of the space described by $\boldsymbol{x}$. In the case of the gradient of the variance, the terms can be assigned by $\tilde{f} = \frac{df}{d\boldsymbol{y}}$ and $\boldsymbol{s} = \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}}$. Using analytic principal sensitivity FOSM, the directional derivative is calculated analytically. This leads to

$$\frac{d\sigma^2{}_f}{d\boldsymbol{y}} = \frac{d^2 f}{d\boldsymbol{y} d\boldsymbol{x}} \cdot \boldsymbol{Cov} \cdot \frac{df}{d\boldsymbol{x}} = \frac{d^2 df}{d\boldsymbol{y} ds} \quad . \tag{39}$$

The resulting equation reveals that analytic PSF is a generalization of the pFOSM approach suggested in [25]. Like for direct differentiation, the resulting second-order direction derivative depends on the problem. The derivation is similar to direct differentiation in chapter 2.1.2. First, one obtains the gradient with respect to design parameters (13) $\frac{df}{d\boldsymbol{y}}$ and some residual equations (14). Afterwards, only one directional derivative, independent of the number of random parameters, is calculated by

$$\frac{d^2 f}{d\boldsymbol{y} ds} = \frac{d^2 f}{d\boldsymbol{y} ds}(\boldsymbol{x}, \boldsymbol{u}, \frac{d\boldsymbol{u}}{ds}) \tag{40}$$

$$\mathbf{0} = \frac{d\boldsymbol{R}}{ds}(\boldsymbol{u}, \boldsymbol{x}, \frac{d\boldsymbol{u}}{ds}) \quad . \tag{41}$$

For instance, the gradient of a linear compliance problem $\frac{df}{dy}$ is given by

$$\frac{df}{d\boldsymbol{y}} = -\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \boldsymbol{u} \tag{42}$$

and the residual equation by

$$\boldsymbol{R} = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f} \quad . \tag{43}$$

Direct differentiation with respect to s leads to

$$\frac{d^2 f}{d\boldsymbol{y} ds} = -2\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \frac{d\boldsymbol{u}}{ds} - \boldsymbol{u}^T \frac{d^2\boldsymbol{K}}{d\boldsymbol{y} ds} \boldsymbol{u} \tag{44}$$

$$\boldsymbol{K} \frac{d\boldsymbol{u}}{ds} = -\frac{d\boldsymbol{K}}{ds}\boldsymbol{u} + \frac{d\boldsymbol{f}}{ds} \quad . \tag{45}$$

The stiffness matrix direction derivatives are calculated using (38) by

$$\frac{d\boldsymbol{K}}{ds} = \sum_i \frac{d\boldsymbol{K}}{dx_i} \cdot s_i \tag{46}$$

$$\frac{d^2\boldsymbol{K}}{d\boldsymbol{y} ds} = \sum_i \frac{d\boldsymbol{K}}{d\boldsymbol{y} dx_i} \cdot s_i \tag{47}$$

with the ith entry s_i of the direction $\boldsymbol{s} = \boldsymbol{Cov}\frac{df}{d\boldsymbol{x}}$. This leads to the overall formula

$$\frac{d\sigma^2}{d\boldsymbol{y}} = 2\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \left[\boldsymbol{K}^{-1} \left[\frac{d\boldsymbol{K}}{ds}\boldsymbol{u} - \frac{d\boldsymbol{f}}{ds} \right] \right] - \boldsymbol{u}^T \frac{d^2\boldsymbol{K}}{d\boldsymbol{y} ds} \boldsymbol{u} \quad . \tag{48}$$

For other objective functions, the derivations and formulas are given in Appendix B.1. Analytic principal sensitivity FOSM delivers the analytic exact gradient of the variance using the same basic concepts of direct differentiation with only one additional system of equations. Therefore, it is without any drawback, superior to direct differentiation. The inverse stiffness matrix in (48) is not computed directly. Instead, the whole product $\boldsymbol{K}^{-1}\left[\frac{d\boldsymbol{K}}{ds}\boldsymbol{u} - \frac{d\boldsymbol{f}}{ds}\right]$ is computed by solving the system of equations. Therefore, the number of systems of equations is the same as by using the adjoint method. Doing some reformulation in the compliance problem shows that the equations are equal, but the equations obtained by aPSF are more efficient and easier to implement, if no reformulations are done. Therefore, a significant difference in calculation time is not expected. However, the adjoint method does not work for eigenfrequency problems [16], since in that case the adjoint system has no solution. As a consequence, the author suggests to use aPSF instead of the adjoint method for calculating the gradient of the variance.

3.3. Overall framework of principal sensitivity FOSM

In the following, implementation frameworks of nPSF and aPSF are suggested. Fig. 3 shows a flow chart for the implementation of nPSF. Here, n is the number of design parameters and m is the number of random parameters. It is assumed that the objective function delivers the function value as well as gradients with respect to random parameters $\frac{df}{d\boldsymbol{x}}$ and with respect to design variables $\frac{df}{d\boldsymbol{y}}$. The inputs of the objective function are the vectors of random variables $\boldsymbol{x}$ and design variables $\boldsymbol{y}$.

Fig. 4 shows a flow chart for the general implementation of aPSF. Here, p describes the number of degrees of freedom. Note that significant details inside the FE-code are missing, since this part is highly intrusive and problem-dependent. The number and the size of internal partial derivatives are dependent on the parametrization, objective function and structure of the FE-code. For instance, for the simple case of compliance minimization, the gradients of the stiffness matrix (given by (46) and (47)) are required.

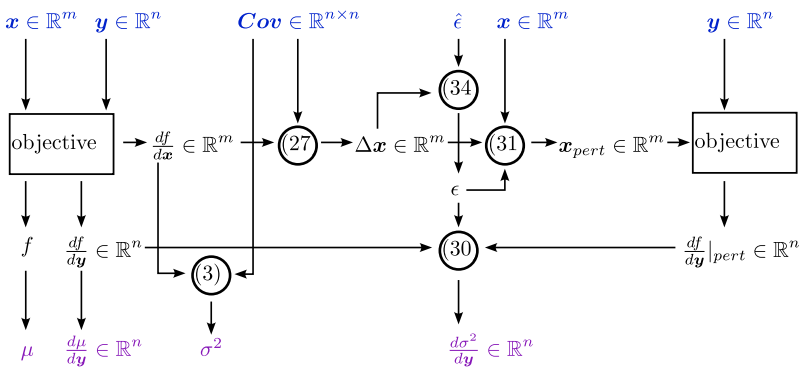


Fig. 3. Flowchart of nPSF. Blue color indicates inputs, purple color indicates outputs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

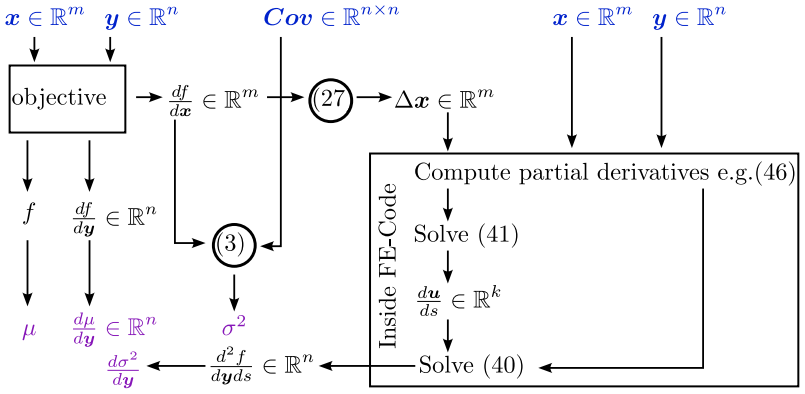


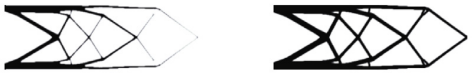
Fig. 4. General flowchart of aPSF. Blue color indicates inputs, purple color indicates outputs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Numerical examples

In the following, both versions of the PSF method are applied to different standard problems for structural optimization. Optimization examples with respect to compliance, stresses, contact problems and principal eigenfrequency are considered. The general optimization problem is defined as

$$\begin{aligned} \min \quad & \bar{f}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \\ \text{s.t.} \quad & \frac{V_0}{V(\boldsymbol{x}, \boldsymbol{y})} - v_{frac} \leq 0 \\ & \mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1} \\ & \mathbf{0} = \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \end{aligned} \tag{49}$$

with the (possibly nonlinear) residual equation $\boldsymbol{R}$, which depends on the deformation vector $\boldsymbol{u}$ and the design variables $\boldsymbol{y}$. The resulting objective function $\bar{f}$ is defined as $\bar{f} = \mu_f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) + \kappa \cdot \sigma_f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$ for a robust optimization and defined as $\bar{f} = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$ for a deterministic optimization. The function f is the problem-dependent objective function. The nPSF method is used as a plugon for the underlying code. Therefore, the implementation effort is very low. In difference, the derivation of aPSF is very time consuming. At the same time, the implementation requires many changes in the underlying FE-code. Therefore, it is only applied to the compliance and the eigenfrequency example.



(a) High η-value → eroded design.

(b) Low η-value → dilated design.

Fig. 5. Influence of globally changed projection threshold parameter η.

All optimizations are done using the Method of moving asymptotes (MMA) by [26] with some modifications by [27]. The move limits are set as initial move limit $s_{ini} = 0.5$, move limit increment $s_{inc} = 1.2$ and move limit decrement $s_{dec} = 0.7$. The optimization problems are parameterized using the SIMP-based topology optimization scheme [28] with a variable filter and projection [29]. The used formulas are given in Appendix A. The FE-analysis is done on in-house tools which use double-precision numbers. In all following examples, with exception of the contact example, random effects are modeled by varying projection threshold parameters η in Eq. (A.2). This framework was proposed the first time by Sigmund in [30] in order to model geometric imperfections. The effect of a globally varying projection threshold parameter is shown in Fig. 5. If the value of η is increased, the whole structure is eroded. Otherwise the design is dilated. In the following examples, the projection threshold parameter η_i is defined for every finite element i individually like described in [9]. This leads to local dilation and erosion. The correlation between the random parameters is computed using an isotropic gaussian random field with a correlation length l_c.

In the following examples, different methods for computing the gradient of the FOSM approximation are compared with respect to accuracy and computation time. General properties of the first-order second-moment method itself are not discussed in detail, since this has already been done in e.g. [14,22]. Therefore, only some results are checked by the Monte-Carlo method. Instead, the results obtained with adjoint based FOSM are mainly used as reference solutions.

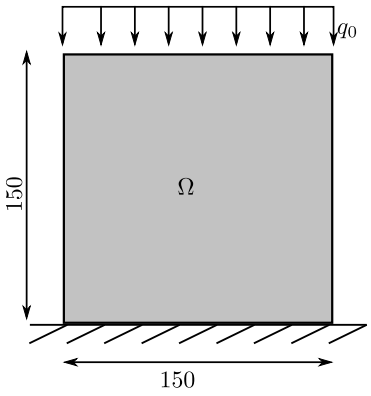
4.1. Compliance topology optimization

Both versions of PSF are used for a compliance topology optimization of a distributed load benchmark like described in [14]. The design space Ω and relevant parameters are summarized in Fig. 6. The optimization problem is defined by (49) and consists of a quadratic building space, which is divided into 22 500 quadrilateral finite elements. The model is clamped on the bottom side and free on the other edges. A distributed load with a load per length of $q_0 = 1$ is applied to the top of the model. As objective function, the compliance defined by

$$f = \boldsymbol{f}^T \boldsymbol{u} \quad , \tag{50}$$

with the load $\boldsymbol{f}$ and the deformation $\boldsymbol{u}$ is considered. The residual is defined by the linear FEM-equation $\boldsymbol{R} = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}$ with the stiffness matrix $\boldsymbol{K}$. In order to model geometric imperfections, the projection threshold parameters η_i of each element i are considered as normal distributed random variables with a standard deviation $\sigma_x = 0.05$. The correlation between these variables is given by an isotropic Gaussian random field with a correlation length $l_c = 20$. For a robust optimization using numeric PSF, no derivations are required, since this method is non-intrusive. For analytic PSF all formulas are derived in (44) to (48).

The deterministic optimization as well as both robust optimizations (aPSF and nPSF) are done using a constant density of $\rho = 0.3$ as an initial design. The number of iterations is restricted to 1500. In order to reach a better local optimum and reduce the nonlinearity of the objective in the first iterations, the projection parameter β is increased by one every 100 iterations. The extremely high number of iterations is chosen to ensure that the design is fully converged in order to have a fair comparison between the methods. Otherwise, small differences might be due to local random effects in convergence behavior and do not represent real differences. However, the changes in the objective function are less then 0.01% per iteration after 10–15 iterations next to a change in β. In Fig. 2 of Section 3.1.2, the error of nPSF is plotted for the initial design at different step sizes and a projection parameter $\beta = 10$. It turns out that the highest accuracy is gained at a normalized step size $\hat{\epsilon} = 10^{-5}$ with a maximum



element size	1
v_{frac}	30%
filter radius	4
load per length q_0	1
correlation length lc	20
projection slope β	increasing 1-10
projection threshold η	0.5
objective κ	5
Young's-modulus	1
Poisson ration	0.3

Fig. 6. Design space and relevant parameters for compliance example.

(a) Deterministic optimized design. (b) Robust optimized design using aPSF/adjoint method. (c) Robust optimized design using nPSF.

Fig. 7. Results of design optimization of distributed load example using different approaches.

relative error of 10^{-5}. Therefore, this step size is chosen for this optimization. The results of the optimizations are shown in Fig. 7. Since the model is similar to the distributed load example in [14], the deterministic optimized design is identical to the reference solution. The robust designs differ from the robust reference solution due to a slightly different distribution of the random variables and a different update strategy for the projection parameter β. Although numeric principal sensitivity FOSM is an approximation of analytic principal sensitivity FOSM, both robust optimizations end up in similar designs. Only small differences are caused by the numeric errors of nPSF. This matches the expectation, since an analysis at the initial design showed that the error of nPSF is very small. The robust designs significantly differ from the deterministic optimized design. Hence, it is obvious that the variance has a major influence on the result of the optimization. The numeric results are shown in Table 1. A Monte Carlo simulation with a sample size of 10'000 is used as a reference. Both FOSM-approaches reach similar results, which have a significantly reduced standard deviation, compared to the deterministic optimization. The first-order second-moment approximation has an low error of 0.5% for the mean value and an moderate error of 15%–25% for the standard deviation.

A major advantage of nPSF compared to other methods is its high efficiency without intrusive behavior. In Fig. 8, the computation time for the gradient of the variance is given for different methods and model sizes. A logarithmic scale is chosen for both axes. All calculations are done on a high performance cluster to ensure that the computation time is represented accurately. Otherwise, a time-dependent system load, rare memory or system cooling issues might lead to significant noise on the computation time. Still, some peaks remain for low number of elements. These peaks are caused by the previously described effects. Finite differences require a large amount of function evaluations. Therefore, the computation time of finite differences is extrapolated from one single function

Table 1
Numeric results of the optimization of the compliance example. The values are obtained by a Monte Carlo simulation with a sample size of 10'000. Results marked with * are obtained with the first-order second-moment method.

approach	μ_f	σ_f
deterministic	92174*, 93323	1614*, 2085
nPSF	95193*, 95612	1075*, 1241
aPSF	95007*, 95445	1066*, 1323

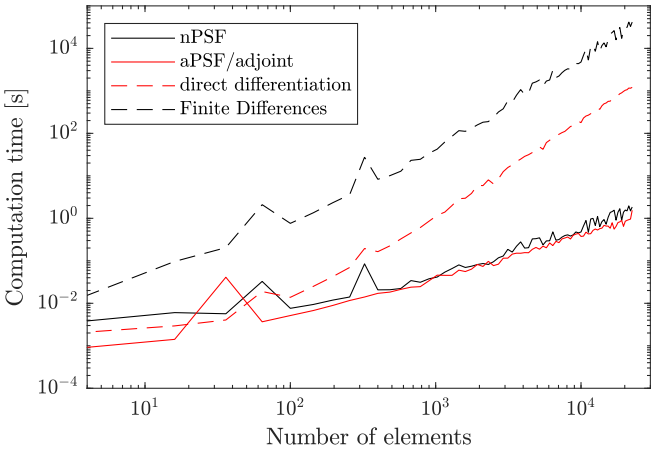


Fig. 8. Computation time for evaluation of the variance gradient for the distributed load example at different numbers of elements.

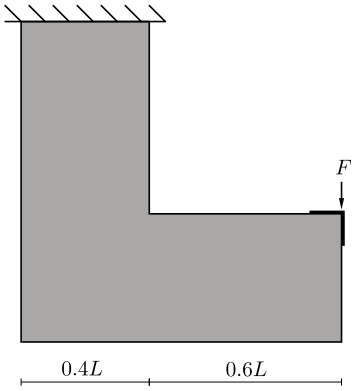
evaluation. The gradient formulations derived using aPSF and the adjoint method are equal. Hence, both methods are represented by the same plot.

According to Fig. 8, aPSF and nPSF have a similar computation time. This matches the expectation, since both methods require only one additional solution of a system of equations of the same size. Differences are caused by bookkeeping and partial derivatives. Compared to direct differentiation and forward finite differences, the complexity is decreased by order one. This is again expected since the number of systems of equations scales with the number of random variables for direct differentiation and finite difference. Thus, nPSF is factor 20,000 faster than finite differences at the given example with 20,000 elements. In every case, the solution of the linear systems of equations takes the major part of the computation time.

In a robust optimization using the first-order second-moment method, the computational cost of one optimization step is determined by the evaluation of the deterministic objective as well as the computation of the gradient of the variance. Previously, only the computation time for the gradient of the variance is compared. Considering a full optimization step, nPSF is still factor 10,000 faster than finite differences. Compared to a deterministic optimization, the computation time using nPSF is only increased by a factor of two.

4.2. Stress-based robust topology optimization

In the following, nPSF is applied to a stress-based robust topology optimization example, showing good gradient approximation quality and big savings in computational cost. For demonstration purpose, the well known stress benchmark example of the L-shaped beam is considered (see e.g. [31]). Analytical PSF is not further pursued here, since gradient derivation lead to the same second order partial derivatives as in [15]. Thus, no improvement nor benefit could be found in using aPSF to approximate the variances gradient of an aggregated stress objective or constraint.



element size	2
load F	10
v_{frac}	40%
filter radius	8
projection slope β	increasing 1-16
projection threshold η	0.5
correlation length lc	30
weighting factor κ	3
Young's-modulus	1
Poisson ration	0.3

Fig. 9. Design space and relevant parameters for L-beam example.

The common L-beam benchmark is depicted in 9. It is discretized with 150 by 150 elements, resulting in 14 400 elements overall. The load F is distributed along the horizontal and vertical edge of the L-beams tip. The loaded edge length is $0.05L$ in both directions, with $L = 300$. Nodes on the top edge are clamped and a Young's modulus of $E = 1.0$ is applied. The Poisson's number is $\nu = 0.3$. The optimization problem is given by (49) with the parameters given in Fig. 9. Again, the residual is defined by $\boldsymbol{R} = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}$ with the stiffness matrix $\boldsymbol{K}$, the deformation vector $\boldsymbol{u}$ and the load vector $\boldsymbol{f}$. As objective function f, the aggregated von Mises stress is considered. The von Mises stress q_i of element i is defined by

$$q_i = \sqrt{\boldsymbol{u}_i^T \boldsymbol{B}_i^T \boldsymbol{C}_i^T \boldsymbol{M} \boldsymbol{C}_i \boldsymbol{B}_i \boldsymbol{u}_i} \tag{51}$$

with the element deformation vector $\boldsymbol{u}_i$, the strain displacement matrix $\boldsymbol{B}_i$, the constitutive matrix $\boldsymbol{C}_i$ and the von Mises matrix $\boldsymbol{M}$ for element i. Aggregation is performed by means of the KS-function [32], given by

$$f = q^{KS} = q_{max} + \frac{1}{\gamma} \log \left[\sum_{i=1}^{n_e} \exp\left(\gamma (q_i - q_{max})\right) \right] \tag{52}$$

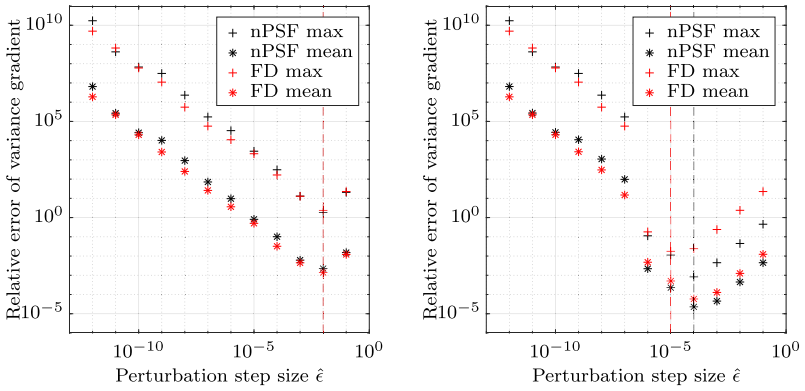
with $q_{max} = \max(\boldsymbol{q})$ and the number of elements n_e. The aggregation parameter γ is updated in each iteration as proposed in [33], such that $\gamma = 15/q_{max}$. To model geometric variations, the projection threshold parameters η_i for every element i are used as uniformly distributed random parameters with a standard deviation $\sigma_x = 0.03$.

As pointed out above, nPSF behaves like finite differences in terms of accuracy with respect to the perturbation step size $\hat{\epsilon}$. First, considering a uniform density field, as it is the initial starting point of an optimization, the accuracy is optimal for a certain step size (see dashed lines in Fig. 10). Nevertheless, the lowest maximum error seems quite high for FD in Fig. 10(a), considering a maximum error in the order of magnitude of 10^0.

In Fig. 11 the relative error is plotted and compared to the order of magnitude of the corresponding variance gradient field. It can be observed, that there is a clear correlation between those two fields. High errors are mostly located in areas with a very low order of magnitude of the gradient. Neglecting all gradient values lower than 10^{-3} results in a drop of the maximum error by approximately three decades (compare Figs. 10(a) and 10(b)).

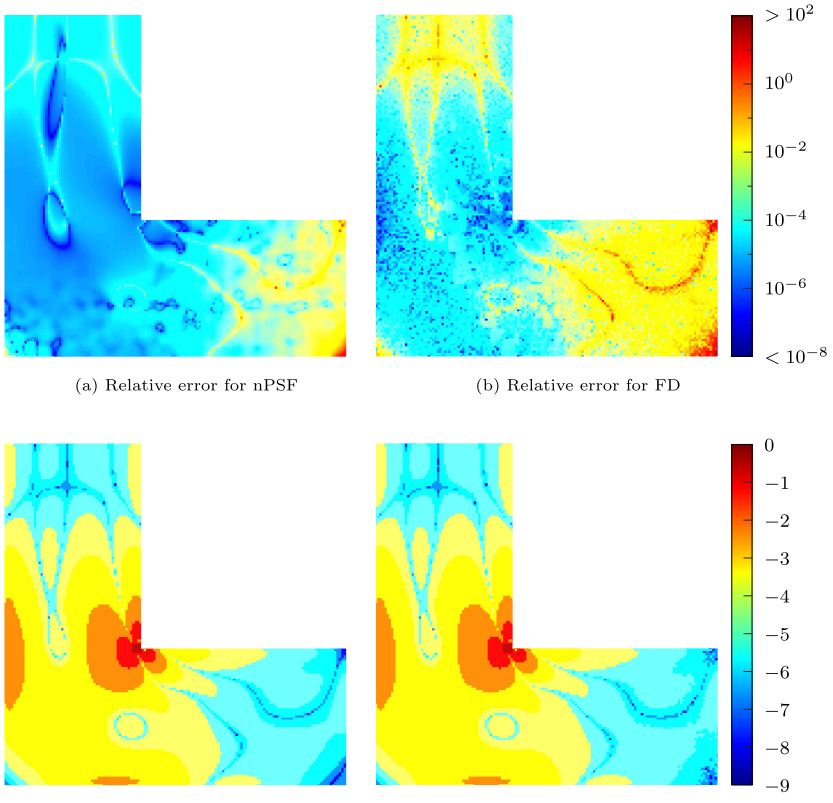
Now, we consider the final iterations of an optimization, where the design variables have settled and the projection parameter β has reached its maximum. An deterministic optimized design variable field is considered. A filter radius of 8 and a projection parameter $\beta = 8$ is applied. Then, performing the same evaluation as before yields Figs. 12 and 13. In contrast to the findings above, the relative error of nPSF is much lower and rather independent of the underlying density field compared to FD. This is expected, since the projection function reliably calculates the gradient in low density areas, whereas for FD high errors occur in those areas. Neglecting all gradient values lower than 10^{-3} results in a big improvement in mean and maximum error for FD. For nPSF the improvements only affect the maximum error.

Comparing the initial and the optimized density fields by only considering the effective filtered gradient, the optimal perturbation step size drops by two orders of magnitude for nPSF, which indicates, that some kind of step



(a) all gradients considered (b) gradients bigger than 10^{-3} considered

Fig. 10. L-beam mean and maximum gradient errors for an uniform density field.

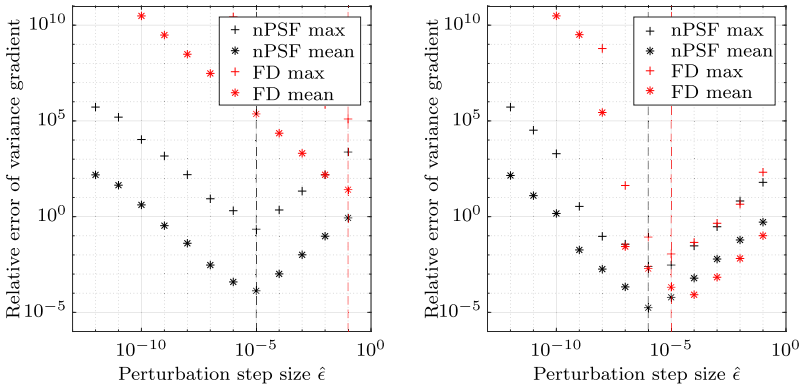


(a) Relative error for nPSF (b) Relative error for FD

(c) Order of Magnitude for nPSF (d) Order of Magnitude for FD

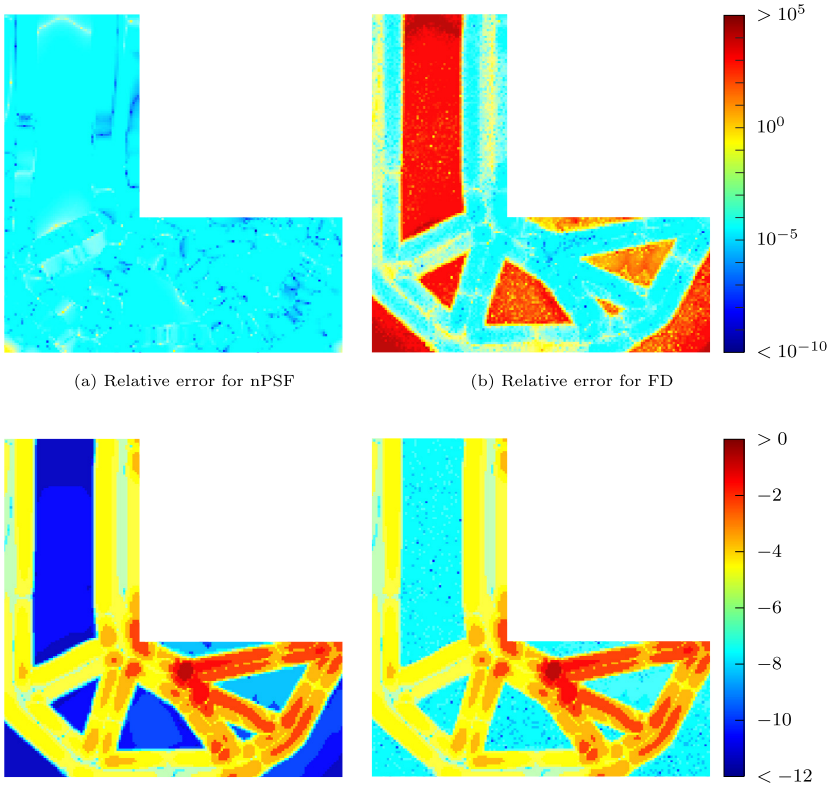
Fig. 11. Local relative error (a–b) and order of magnitude (c–d) of the variance gradient for an uniform density field. Both plotted for nPSF (a and c) and FD (b and d) at the corresponding optimal perturbation step size $\hat{\epsilon}$.

size adaptation during optimization might be reasonable. This is not covered within this work and is left to future research.



(a) All gradients considered (b) Gradients bigger than 10^{-3} considered

Fig. 12. L-beam mean and maximum gradient errors for an optimized density field.

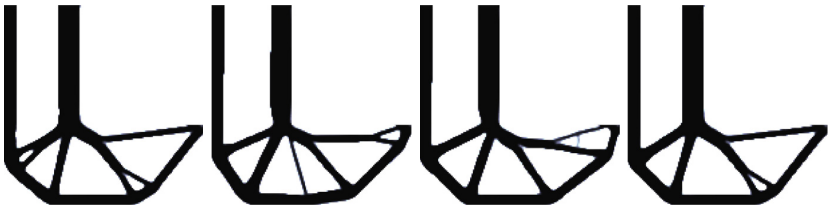


(a) Relative error for nPSF (b) Relative error for FD

(c) Order of Magnitude for nPSF (d) Order of Magnitude for FD

Fig. 13. Local relative error and order of magnitude for nPSF and FD at the optimal perturbation step size $\hat{\epsilon}$ and optimized density field.

Evaluating nPSF optimized designs for different perturbation step sizes and comparing to adjoint based FOSM (see Fig. 14) shows a constant mean value (±2%). Furthermore, a similar design is obtained if the step size $\hat{\epsilon}$ is chosen optimal with respect to Fig. 12(b) (compare Figs. 14(a) and 14(b)). This is expected, since the mean value is dominating the objective function value and the gradients are comparable for both approaches. The standard



(a) Adjoint based FOSM
$\mu_f = 2.067$
$\sigma_f = 1.34 \cdot 10^{-3}$

(b) nPSF ($\hat{\epsilon} = 10^{-2}$)
$\mu_f = 2.110$
$\sigma_f = 1.59 \cdot 10^{-3}$

(c) nPSF ($\hat{\epsilon} = 10^{-4}$)
$\mu_f = 2.118$
$\sigma_f = 1.28 \cdot 10^{-3}$

(d) nPSF ($\hat{\epsilon} = 10^{-6}$)
$\mu_f = 2.066$
$\sigma_f = 1.51 \cdot 10^{-3}$

Fig. 14. Optimized L-beam designs. Values are obtained using the first-order second-moment method.

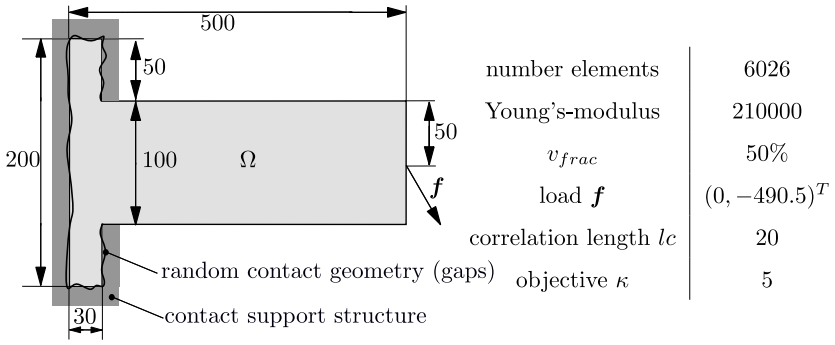


Fig. 15. Design space and relevant parameters for the contact optimization example.

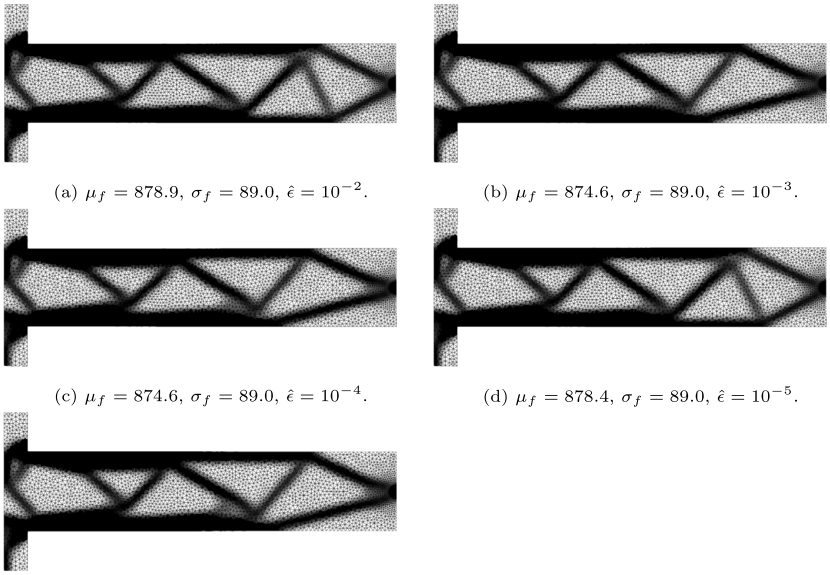
deviation only adds little, even with a weighting factor $\kappa = 3$. Thus, a deviation of $\pm 20\%$ appears reasonable and can be explained by different local minimal the optimizer is lead to.

In case of an aggregated stress function, the computational savings are significant, if the variance gradient is approximated by nPSF. To quantify the improved computational time, the average time needed for one iteration is compared. From a theoretical point of view, the savings are enormous. Numerical PSF only requires one additional equation solve and function evaluation to calculate the variances gradient. By contrast, adjoint-based FOSM requires two equation solves, the computation of several second-order partial derivatives and corresponding matrix operations on those full and non-sparse matrices (for details see [15]). In this example, the computation time differs by approximately a factor of ten.

4.3. Topology optimization of contact problems

The nPSF is applied to contact-constrained topology optimization, similar to [34], where the compliance is minimized while the geometry of contact support is scattered randomly. The used robust topology optimization algorithms are based on the general framework for topology optimization of structures in unilateral contact by Stromberg and Klarbring [35]. In doing so, an augmented Lagrangian approach is formulated to incorporate Signorini's contact conditions into the equilibrium equations [36]. This leads to the optimization problem (49) with the non-linear contact equilibrium equation as residual equation $\boldsymbol{R}$. The objective function f is the compliance given by $f = \boldsymbol{f}^T \boldsymbol{u}$ with the external load $\boldsymbol{f}$ and the deformation $\boldsymbol{u}$. The contact equilibrium equation is discontinuous regarding the contact geometry. However, it is continuous regarding the design variables. Therefore, gradient-based optimization methods like MMA can be used. The required gradients of the compliance are provided in [34].

The considered example is shown in Fig. 15. It consists of a cantilever, which is discretized by 6026 triangular finite elements. A vertical load is applied to the middle node of the right edge. Additional contact support structures



(a) $\mu_f = 878.9$, $\sigma_f = 89.0$, $\hat{\epsilon} = 10^{-2}$.

(b) $\mu_f = 874.6$, $\sigma_f = 89.0$, $\hat{\epsilon} = 10^{-3}$.

(c) $\mu_f = 874.6$, $\sigma_f = 89.0$, $\hat{\epsilon} = 10^{-4}$.

(d) $\mu_f = 878.4$, $\sigma_f = 89.0$, $\hat{\epsilon} = 10^{-5}$.

(e) $\mu_f = 874.6$, $\sigma_f = 89.0$, adjoint based FOSM.

Fig. 16. Optimized domains using nPSF with different perturbations $\hat{\epsilon}$ and FOSM. Values are obtained by the first-order second-moment method.

are used to apply the concept of form closure at the flange. Imperfections are modeled by a random contact geometry. The node-coordinates of the flange, which are perpendicular to the contact support structure, are assumed to be normal distributed random variables with a standard deviation $\sigma_x = 0.03$. Correlation is modeled by an isotropic exponential correlation function with a correlation length $l_c = 25$.

Figs. 16(a) to 16(d) show the optimized domains using the nPSF based gradient computation for different perturbation step-sizes, whereas Fig. 16(e) is optimized with the adjoint based FOSM gradient computation. The optimized domains of Figs. 16(b), 16(c) and the reference design of Fig. 16(e) show, that the same optimized designs are computed if the perturbation step-size is chosen carefully. However, the four strut design is not found after 1000 iterations, if the perturbation is chosen too large or too small, as shown in Figs. 16(a) and 16(d). In consequence, their means of the compliance are approx. 0.5% higher than the mean of the reference optimized design, using adjoint based FOSM.

Additionally, the relative error of nPSF in comparison to adjoint based FOSM is visualized in Fig. 17. The relative error is computed based on the derivative of the standard deviation using nPSF and the adjoint based FOSM.

As already shown in the other examples, the accuracy of the nPSF does strongly depend on the chosen perturbation step-size $\hat{\epsilon}$. In contrast to the other examples, the course of the relative error is non-continuous. It turns out, that the relative error has a jump between the perturbation parameters $\hat{\epsilon} = 0.001$ and $\hat{\epsilon} = 0.01$. This happens because nodes at the contact surface lift off from the support structure, if too large perturbations of the random variables are applied. In consequence, the non-smooth equilibrium equations of the contact problem change in a non-continuous manner. Therefore, it is to be expected that the gradient of nPSF does change non-continuously as well. In this particular case, a contact node of the upper right vertical support structure lifts off and loses the contact to the support. This phenomenon depends on the chosen discretization. Thus, it might happen at a different node at the support and for a different perturbation step-size, if a different discretization is used. Considering these observations it is hard to choose an appropriate step-size. In order to reduce the risk of discontinuous jumps, the step-size should be chosen as small as possible, at the cost of increased rounding errors. Therefore, a step-size of $\hat{\epsilon} = 10^{-4}$ seems to be a good compromise, which is supported by the optimization results. If the gradient is computed using adjoint based FOSM, the problem of discontinuous jumps does not occur, since the non-smooth equilibrium equations are evaluated only for the mean values of the random parameters. The issue could only occur, if the gradient is evaluated exactly on the discontinuity, which is very unlikely.

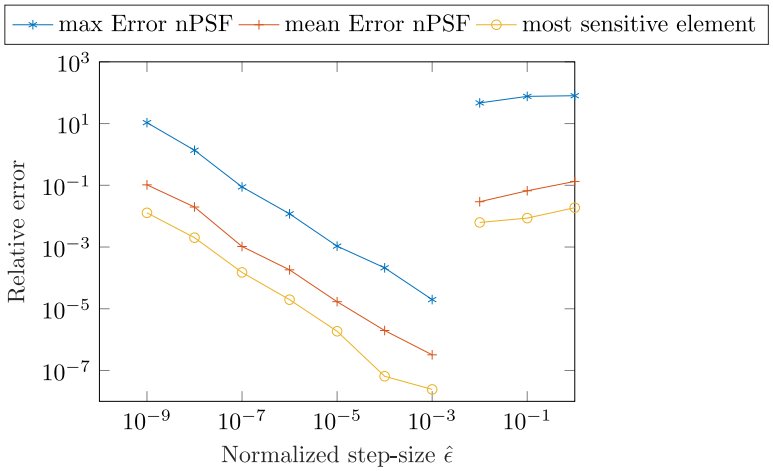


Fig. 17. Relative Error of numeric principal sensitivity FOSM (nPSF) with respect to adjoint based FOSM of the contact example for different step-sizes.

The computation time using nPSF instead of adjoint-based FOSM increases significantly, which is expected for this example. Using adjoint-based FOSM, an additional system of linear equations of the same dimensions as the equilibrium equations must be solved, whereas using nPSF the non-linear equilibrium equations must be solved for a different set of the random variables. A line-search algorithm is used to solve the contact problem so that the non-linear solver must solve several additional linear systems of the same size until converges is observed. Additionally, the perturbed random geometry can lead to poor convergence behavior of the non-linear solver, since Signorini's contact conditions enforce that the displacements at the support must not penetrate into the perturbed support geometry. In consequence, large perturbations can change the convergence behavior of the non-linear contact problem drastically, so that even higher costs occur, especially if the perturbation step-size is large. In the presented numerical example, a normalized step-size of $\hat{\epsilon} = 0.0001$ is twice as fast, than a step-size of $\hat{\epsilon} = 0.1$. Nonetheless, the computation time using the "faster" perturbation step-size of $\hat{\epsilon} = 0.0001$ is still twice as high as if adjoint-based FOSM is used. This is expected, since adjoint based FOSM does not solve an additional contact problem.

4.4. Topology optimization of the principal eigenfrequency

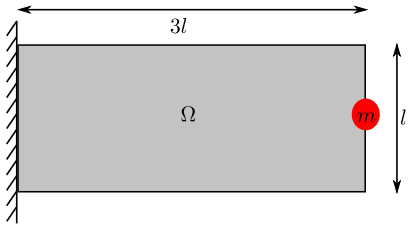
In a last example, both versions of PSF are applied to a topology optimization of the principal eigenfrequency similar to [37]. A robust optimization using nPSF to this example has been already done in [16]. The model consists of a homogeneous cantilever beam with the density $\rho_0 = 1$, which is clamped on the left side. A lumped mass with the value $m = 3$ is added to the middle element on the right edge. The design space Ω and relevant parameters are summarized in 18. The optimization problem is defined by (49) with the principal eigenfrequency as objective function. Since eigenfrequency problems are prone to mode switching, the principal eigenfrequency is approximated by the p-norm

$$f = \left[\sum_{j=1}^{n} \omega_j^p \right]^{1/p} , \tag{53}$$

where ω_i is defined as the ith eigenfrequency. This smooth approximation has been proposed and proven in [38]. The residual equation is defined as the eigenvalue problem

$$\boldsymbol{R} = \left[\boldsymbol{K} - \omega_j^2 \boldsymbol{M} \right] \boldsymbol{\varphi}_j = \boldsymbol{0} \tag{54}$$

with the stiffness matrix $\boldsymbol{K}$, the mass matrix $\boldsymbol{M}$ and the corresponding eigenvector $\boldsymbol{\varphi}_j$. The aggregation function should converge to the minimum eigenfrequency and induce as few nonlinearities as possible. Therefore, the



l	130
element size	1
v_{frac}	25%
m	3
Filter radius	3
Correlation length l_c	40
Projection slope β	increasing 1-10
Projection threshold η	0.5
objective κ	5
Youngs-modulus	1
Poisson ration	0.3
Density ρ_0	1

Fig. 18. Design space and parameters of the eigenfrequency example.

parameter p is chosen as $p = -5$. In order to model geometric imperfections, the projection threshold parameters η_i for every element i are used as normal distributed random parameters with a standard deviation $\sigma_x = 0.05$. Correlation is modeled using a Gaussian random field with a correlation length $l_c = 40$.

In [16] it is described that the adjoint method does not work for calculating the gradient of the variance at eigenvalue problems. The reason is an adjoint system which has no solution. In difference, nPSF works for every objective function as long as the gradient exists. As an analytic alternative, aPSF is derived for eigenvalue problems in Appendix B.2. Using this method, only one additional systems of equations has to be solved. Since the p-norm of the three principal eigenvalues is considered as objective function, the gradient formula has to be modified like described in Appendix B.3. This results in one additional system of equations for every considered eigenvalue. All analytic formulas are validated using finite differences.

All optimizations use a constant density of $\rho = 0.25$ as initial design. The number of iterations is restricted to 1500. Again, this extremely high value is chosen to ensure that the design is fully converged. In order to reach a better local optimum and reduce the nonlinearity of the objective, the projection parameter β is increased by one every 100 iterations. Fig. 19 shows the accuracy of nPSF for different normalized step sizes $\hat{\epsilon}$ at the initial design. The left plot shows the mean and the maximum relative error of the gradient estimation. The best performance is reached at a normalized step size of $\hat{\epsilon} = 10^{-4}$ with a mean error of 10^{-5}. Hence, this step size is chosen for the following optimization. However, the maximum error reaches values of up to 10%. Similar to the stress example, the error is high for gradient entries near to zero. These elements only have low influence on the optimization. In the right plot, all entries which are factor 10^{-4} smaller than the maximum gradient entry are removed. Hence, the maximum error is reduced to less than 0.1%.

The optimizations lead to the designs shown in Fig. 20. The numeric results are shown in Table 2. It can be seen that the deterministic optimized design is equal to the result in [37]. However, it is different to [16] because the referred paper uses another optimization algorithm and reaches another local minimum. The differences between the designs are highlighted in Fig. 21. Black elements show increased density, while white elements show decreased density values. Grey space indicate no difference between the designs. Both robust optimized designs are similar and reach the same numeric results. Hence, the error in nPSF has no influence on the result. At the same time, they significantly differ from the deterministic optimized design. Therefore, the variance (gradient) has influence on the optimization results.

Since the adjoint method does not work for this method, savings in computational cost are significant. Fig. 22 shows the computation time for the computation of the gradient of the variance for different numbers of elements in a double logarithmic plot. For finite differences, the computation time is again extrapolated from one function evaluation. Both versions of PSF are one complexity order faster than direct differentiation and finite differences. At

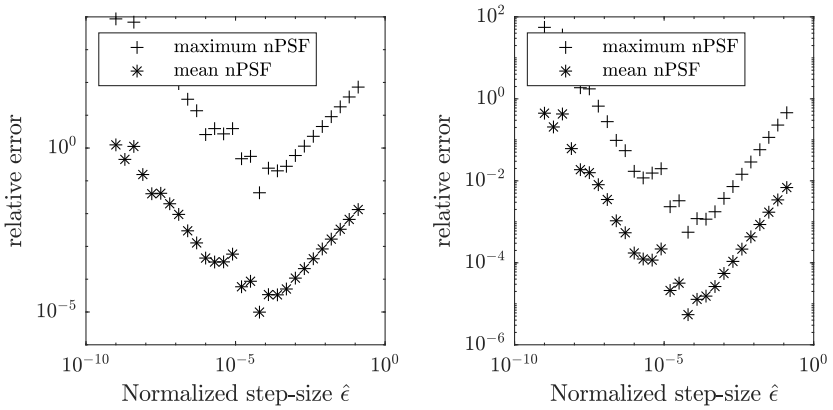
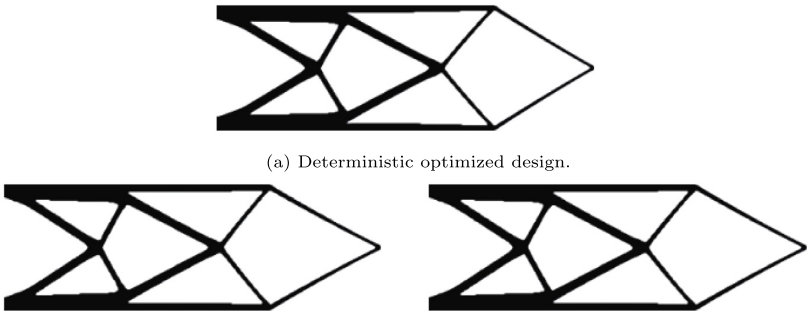


Fig. 19. Accuracy of numeric principal sensitivity FOSM at eigenfrequency example for different step sizes. Left: Unfiltered Relative error, Right: filtered relative error. For filtering, all gradient entries which are three decades smaller than maximum entry are removed.

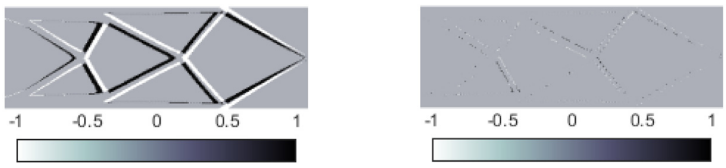


(a) Deterministic optimized design.

(b) Robust optimized design using aPSF.

(c) Robust optimized design using nPSF.

Fig. 20. Results of the optimizations with respect to the principal eigenfrequency.



(a) Difference between deterministic optimized design and robust optimized design.

(b) Difference between robust optimized design using aPSF and nPSF.

Fig. 21. Differences between designs of eigenfrequency example. Black elements indicate increased density, white elements indicate decreased density.

the given model, the difference compared to direct differentiation is a factor of order 1000. The computation time of aPSF and nPSF is similar, although the solution of one eigenvalue problem is often more expensive than the solution of a system of equations of the same size. A reason is given by the effect of the p-norm. Since three eigenfrequencies are considered, all parts of aPSF, including systems of equations, bookkeeping and matrix derivatives have to be

Table 2
Numeric results of the optimization of the eigenfrequency example. The values are obtained by a Monte Carlo simulation with a sample size of 10'000. Results marked with * are obtained with the first-order second-moment method.

approach	μ_f	σ_f
deterministic	699.6*, 694.1	5.1*, 6.3
nPSF	698.4*, 694.1	3.7*, 5.2
aPSF	698.4*, 694.1	3.7*, 5.2

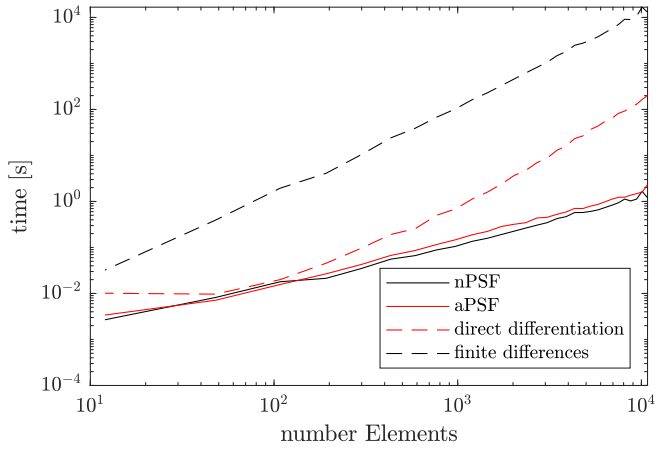


Fig. 22. Computation time for computing the gradient of the variance for the eigenfrequency example at different model sizes.

done three times. In difference, nPSF only needs one additional function evaluation, independent of the number of considered eigenfrequencies.

5. Conclusions and outlook

A new robust design optimization framework is presented, which is based on the first-order second-moment method. The difference to other approaches is given by the computation of the gradient of the variance. While some existing methods, such as direct differentiation and adjoint formulations, are very problem-dependent and intrusive, other methods, such as finite differences, are computational costly. The new proposed approach comes in two versions and is called principal sensitivity first-order second-moment method (PSF). It is based on the directional derivative of the objective gradient.

The numeric version of PSF only requires one additional evaluation of the objective gradient. Therefore, it does not need access to the FE-Code and can be used as a plugon to commercial code. This reduces the required memory significantly. At the same time, it is as fast as the adjoint method for many examples. In the worst case, the computation time is only doubled compared to the adjoint method. Similar to finite differences, a step-size has to be chosen carefully. Hence, a procedure for determining a problem-independent step size is suggested.

The analytic version of PSF computes the directional derivative analytically. Therefore, it is exact but intrusive. For several objective functions, it leads to the same formulas as the adjoint method. However, for the example of the principal eigenfrequency, the adjoint method does not work while analytic PSF leads to a correct gradient of the variance.

The provided approach is successfully applied to several benchmark examples.

In future work, this method will applied to several examples from different disciplines in science, since is universally applicable. At the same time, methods for adaptive step-size control should be derived to the principal sensitivity first-order second-moment method in order to guarantee its accuracy and make it applicable in industrial use cases.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

Financial support of the German Research Foundation (reference number KR 4914/13-1) is acknowledged. Parts of the current paper have been conducted in the research training group "Simulation-Based Design Optimization of Dynamic Systems Under Uncertainties" (SENSUS) funded by the state of Hamburg within the Landesforschungsförderung, Germany under project number LFF-GK11.

Appendix A. Topology optimization formulas.

In all numerical examples, the design variables are filtered using a variable filter. The filtered densities $\hat{\rho}_e$ are given by

$$\hat{\rho}_e = \frac{\sum w_{ei} y_i}{\sum w_{ei}} \tag{A.1}$$

with $w_{ei} = \max(0, R - r_{ei})$, the design variable y_i and the filter radius R. Here, r_{ei} describes the distance between elements i and j. The filtered densities are projected using a Heaviside approximation described by

$$\rho_i = \frac{\tanh(\beta\eta_i) + \tanh(\beta\hat{\rho} - \beta\eta_i)}{\tanh(\beta\eta_i) + \tanh(\beta - \beta\eta_i)} \tag{A.2}$$

with the threshold parameter η_i of element i and the slope parameter β. Using the SIMP scheme, the element E-modulus is computed by

$$E_i = E_0 \cdot \rho_i^p \quad , \tag{A.3}$$

with the E-modulus of a dense element E_0 and the SIMP parameter p. In this paper, the SIMP parameter is chosen as $p = 3$. The element mass is computed analogous.

Appendix B. Analytic PSF for different problems.

B.1. Analytic PSF for deformation-vector entries of static linear problem

Problem described by

$$f = \boldsymbol{e}^T \boldsymbol{u} \tag{B.1}$$

with derivative

$$\frac{df}{d\boldsymbol{y}} = \boldsymbol{\lambda}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \boldsymbol{u} \tag{B.2}$$

$$\boldsymbol{0} = \boldsymbol{K}\boldsymbol{\lambda} + \boldsymbol{e} \tag{B.3}$$

$$\boldsymbol{0} = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f} \tag{B.4}$$

differentiation leads to

$$\frac{d^2 f}{d\boldsymbol{y} ds} = \frac{d\boldsymbol{\lambda}^T}{ds} \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \boldsymbol{u} + \boldsymbol{\lambda}^T \frac{d^2\boldsymbol{K}}{d\boldsymbol{y} ds} \boldsymbol{u} + \boldsymbol{\lambda}^T \frac{d\boldsymbol{K}}{d\boldsymbol{y}} \frac{d\boldsymbol{u}}{ds} \tag{B.5}$$

$$\boldsymbol{0} = \frac{d\boldsymbol{K}}{ds}\boldsymbol{\lambda} + \boldsymbol{K}\frac{d\boldsymbol{\lambda}}{ds} + \frac{d\boldsymbol{e}}{ds} \tag{B.6}$$

$$\boldsymbol{0} = \frac{d\boldsymbol{K}}{ds}\boldsymbol{u} + \boldsymbol{K}\frac{d\boldsymbol{u}}{ds} - \frac{d\boldsymbol{f}}{ds} \tag{B.7}$$

Solve two additional systems of equations

B.2. Analytic PSF for principal eigenfrequency of linear problems.

The corresponding eigenvalue problem is given by

$$\left[\boldsymbol{K} - \omega^2 \boldsymbol{M}\right] \boldsymbol{\varphi} = \boldsymbol{0} \tag{B.8}$$

with the derivative [39]

$$\frac{d\omega^2}{d\boldsymbol{y}} = \boldsymbol{\varphi}^T \left[\frac{d\boldsymbol{K}}{d\boldsymbol{y}} - \omega^2 \frac{d\boldsymbol{M}}{d\boldsymbol{y}}\right] \boldsymbol{\varphi} \quad . \tag{B.9}$$

Differentiating with respect to principal sensitivity direction $\boldsymbol{s}$ leads to

$$\frac{d^2\omega^2}{d\boldsymbol{y}ds} = 2\boldsymbol{\varphi}^T \left[\frac{d\boldsymbol{K}}{d\boldsymbol{y}} - \omega^2 \frac{d\boldsymbol{M}}{d\boldsymbol{y}}\right] \frac{d\boldsymbol{\varphi}}{ds} + \boldsymbol{\varphi}^T \left[\frac{d^2\boldsymbol{K}}{d\boldsymbol{y}ds} - \frac{d\omega^2}{ds}\frac{d\boldsymbol{M}}{d\boldsymbol{y}} - \omega^2 \frac{d^2\boldsymbol{M}}{d\boldsymbol{y}ds}\right] \boldsymbol{\varphi} \tag{B.10}$$

with

$$\left[\boldsymbol{K} - \omega^2 \boldsymbol{M}\right] \frac{d\boldsymbol{\varphi}}{ds} + \left[\frac{d\boldsymbol{K}}{ds} - \frac{d\omega^2}{ds}\boldsymbol{M} - \omega^2 \frac{d\boldsymbol{M}}{ds}\right] \boldsymbol{\varphi} = \boldsymbol{0} \tag{B.11}$$

The problem is solved using Nelsons formula [40]

B.3. Analytic PSF for P-norm of objective function

The P-norm f_p of a data set $\{f_1, f_2, \ldots, f_n\}$ is given by

$$f_p = \left[\sum_{i=1}^{n} f_i^p\right]^{1/p} \tag{B.12}$$

with an arbitrarily chosen exponent p. Differentiation with respect to design variables $\boldsymbol{y}$ leads to

$$\frac{df_p}{d\boldsymbol{y}} = \left[\sum_{i=1}^{n} f_i^p\right]^{1/p-1} \cdot \left[\sum_{i=1}^{n} f_i^{p-1} \cdot \frac{df_i}{d\boldsymbol{y}}\right] \quad . \tag{B.13}$$

Further differentiating with respect to principal sensitivity direction $\boldsymbol{s}$ leads to

$$\begin{aligned} \frac{d^2 f_p}{d\boldsymbol{y}ds} = & \left[\sum_{i=1}^{n} f_i^p\right]^{1/p-1} \cdot \left[\sum_{i=1}^{n} f_i^{p-1} \cdot \frac{d^2 f_i}{d\boldsymbol{y}ds}\right] \\ & + \left[\sum_{i=1}^{n} f_i^p\right]^{1/p-1} \cdot \left[\sum_{i=1}^{n} f_i^{p-2} \cdot (p-1) \cdot \frac{df_i}{ds} \cdot \frac{df_i}{d\boldsymbol{y}}\right] \\ & + (1-p) \cdot \left[\sum_{i=1}^{n} f_i^p\right]^{1/p-2} \cdot \left[\sum_{i=1}^{n} f_i^{p-1} \cdot \frac{df_i}{ds}\right] \cdot \left[\sum_{i=1}^{n} f_i^{p-1} \cdot \frac{df_i}{d\boldsymbol{y}}\right] \end{aligned} \tag{B.14}$$

References

[1] G.-J. Park, T.-H. Lee, K.H. Lee, K.-H. Hwang, Robust design: An overview, AIAA J. 44 (1) (2006) 181–191, http://dx.doi.org/10.2514/1.13639.

[2] G.I. Schuëller, H.A. Jensen, Computational methods in optimization considering uncertainties – An overview, Comput. Methods Appl. Mech. Engrg. 198 (1) (2008) 2–13, http://dx.doi.org/10.1016/j.cma.2008.05.004, URL http://www.sciencedirect.com/science/article/pii/S0045782508002028.

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

[4] T. Chatterjee, S. Chakraborty, R. Chowdhury, A critical review of surrogate assisted robust design optimization, Arch. Comput. Methods Eng. 26 (1) (2019) 245–274, http://dx.doi.org/10.1007/s11831-017-9240-5, URL http://link.springer.com/10.1007/s11831-017-9240-5.

[5] O. Sigmund, On the usefulness of non-gradient approaches in topology optimization, Struct. Multidiscip. Optim. 43 (5) (2011) 589–596, http://dx.doi.org/10.1007/s00158-011-0638-7.

[6] K.-r. Bae, S. Wang, K.K. Choi, Reliability-based topology optimization, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, pp. AIAA 2002–5542, Atlanta, Georgia, USA, 2002, http://dx.doi.org/10.2514/6.2002-5542.

[7] K. Maute, D.M. Frangopol, Reliability-based design of MEMS mechanisms by topology optimization, Comput. Struct. 81 (8) (2003) 813–824, http://dx.doi.org/10.1016/S0045-7949(03)00008-7, URL http://www.sciencedirect.com/science/article/pii/S0045794903000087.

[8] H.-S. Jung, S. Cho, Reliability-based topology optimization of geometrically nonlinear structures with loading and material uncertainties, Finite Elem. Anal. Des. 41 (3) (2004) 311–331, http://dx.doi.org/10.1016/j.finel.2004.06.002, URL http://linkinghub.elsevier.com/retrieve/pii/S0168874X0400112X.

[9] M. Schevenels, B.S. Lazarov, O. Sigmund, Robust topology optimization accounting for spatially varying manufacturing errors, Comput. Methods Appl. Mech. Engrg. 200 (49–52) (2011) 3613–3627, http://dx.doi.org/10.1016/j.cma.2011.08.006, URL http://www.sciencedirect.com/science/article/pii/S0045782511002611.

[10] B.S. Lazarov, M. Schevenels, O. Sigmund, Topology optimization with geometric uncertainties by perturbation techniques, Internat. J. Numer. Methods Engrg. 90 (11) (2012) 1321–1336, http://dx.doi.org/10.1002/nme.3361, URL http://onlinelibrary.wiley.com/doi/10.1002/nme.3361/abstract.

[11] V. Keshavarzzadeh, F. Fernandez, D.A. Tortorelli, Topology optimization under uncertainty via non-intrusive polynomial chaos expansion, Comput. Methods Appl. Mech. Engrg. 318 (2017) 120–147, http://dx.doi.org/10.1016/j.cma.2017.01.019, URL http://linkinghub.elsevier.com/retrieve/pii/S0045782516313019.

[12] I. Doltsinis, Z. Kang, Robust design of structures using optimization methods, Comput. Methods Appl. Mech. Engrg. 193 (23) (2004) 2221–2237, http://dx.doi.org/10.1016/j.cma.2003.12.055, URL http://www.sciencedirect.com/science/article/pii/S0045782504000787.

[13] I. Doltsinis, Z. Kang, G. Cheng, Robust design of non-linear structures using optimization methods, Comput. Methods Appl. Mech. Engrg. 194 (12) (2005) 1779–1795, http://dx.doi.org/10.1016/j.cma.2004.02.027, URL http://www.sciencedirect.com/science/article/pii/S0045782504004165.

[14] B. Kriegesmann, J.K. Lüdeker, Robust compliance topology optimization using the first-order second-moment method, Struct. Multidiscip. Optim. 60 (1) (2019) 269–286, http://dx.doi.org/10.1007/s00158-019-02216-8.

[15] M. Kranz, J. Lüdeker, B. Kriegesmann, A generalized approach for robust topology optimization using the first-order second-moment method for arbitrary response functions, Struct. Multidiscip. Optim. 66 (98) (2023) http://dx.doi.org/10.1007/s00158-023-03540-w.

[16] J.C. Krüger, B. Kriegesmann, Efficient robust topology optimization of eigenfrequencies using the first-order second-moment method, in: Proceedings of the IUTAM Symposium on Optimal Design and Control of Multibody Systems - Adjoint Methods, Alternatives, and Beyond (July 18-21, 2022), 2023, in press.

[17] K. Steltner, C.B.W. Pedersen, B. Kriegesmann, Semi-intrusive approach for stiffness and strength topology optimization under uncertainty, Opt. Eng. (2022) http://dx.doi.org/10.1007/s11081-022-09770-z.

[18] K. Steltner, B. Kriegesmann, C.B. Pedersen, Robust sizing optimization of stiffened panels subject to geometric imperfections using fully nonlinear postbuckling analyses, Thin-Walled Struct. 175 (2022) 109195, http://dx.doi.org/10.1016/j.tws.2022.109195, URL https://www.sciencedirect.com/science/article/pii/S0263823122001719.

[19] C.A. Cornell, Structural safety specifications based on second-moment reliability analysis, IABSE Reports of the Working Commissions, 4, 1969, pp. 235–245, http://dx.doi.org/10.5169/seals-5948, URL https://www.e-periodica.ch/digbib/view?pid=bse-re-001:1969:4::324.

[20] B. Kriegesmann, Robust design optimization with design-dependent random input variables, Struct. Multidiscip. Optim. 61 (2) (2020) 661–674, http://dx.doi.org/10.1007/s00158-019-02388-3.

[21] P. Michaleris, D.A. Tortorelli, C.A. Vidal, Tangent operators and design sensitivity formulations for transient non-linear coupled problems with applications to elastoplasticity, Internat. J. Numer. Methods Engrg. 37 (14) (1994) 2471–2499, http://dx.doi.org/10.1002/nme.1620371408, URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620371408/abstract.

[22] B. Kriegesmann, On the applicability of first-order approximations for design optimization under uncertainty, in: N. Challamel, J. Kaplunov, I. Takewaki (Eds.), Modern Trends in Structural and Solid Mechanics 3 - Non-Deterministic Mechanics, John Wiley & Sons, Ltd, 2021, pp. 39–60, http://dx.doi.org/10.1002/9781119831839.ch3, Section: 3 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119831839.ch3, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119831839.ch3.

[23] H. Møller, Analysis and Optimization for Fluid-Structure Interaction Problems (Ph.D. thesis), in: Special Report / Institute of Mechanical Engineering, (47) Aalborg Universitet, 2002.

[24] W. Squire, G. Trapp, Using complex variables to estimate derivatives of real functions, SIAM Rev. 40 (1) (1998) 110–112, http://dx.doi.org/10.1137/S003614459631241X, URL http://epubs.siam.org/doi/10.1137/S003614459631241X.

[25] K.B. Fragkos, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou, pFOSM: An efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products, Comput. & Fluids 181 (2019) 57–66, http://dx.doi.org/10.1016/j.compfluid.2019.01.016, URL http://www.sciencedirect.com/science/article/pii/S0045793018307680.

[26] K. Svanberg, The method of moving asymptotes - a new method for structural optimization, Internat. J. Numer. Methods Engrg. 24 (2) (1987) 359–373, http://dx.doi.org/10.1002/nme.1620240207, URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620240207/abstract.

[27] L. Li, K. Khandelwal, Two-point gradient-based MMA (TGMMA) algorithm for topology optimization, Comput. Struct. 131 (2014) 34–45, http://dx.doi.org/10.1016/j.compstruc.2013.10.010, URL http://www.sciencedirect.com/science/article/pii/S0045794913002812.

[28] M.P. Bendsøe, O. Sigmund, Topology Optimization Theory, Methods, and Applications, second ed., Springer-Verlag Berlin Heidelberg, 2004, URL http://www.springer.com/us/book/9783540429920#aboutBook.

[29] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Struct. Multidiscip. Optim. 43 (6) (2011) 767–784, http://dx.doi.org/10.1007/s00158-010-0602-y.

[30] O. Sigmund, Manufacturing tolerant topology optimization, Acta Mech. Sinica 25 (2) (2009) 227–239, http://dx.doi.org/10.1007/s10409-009-0240-z, URL https://link.springer.com/article/10.1007/s10409-009-0240-z.

[31] G.A. da Silva, E.L. Cardoso, Stress-based topology optimization of continuum structures under uncertainties, Comput. Methods Appl. Mech. Engrg. 313 (2017) 647–672, http://dx.doi.org/10.1016/j.cma.2016.09.049, URL http://www.sciencedirect.com/science/article/pii/S0045782516312816.

[32] G. Kreisselmeier, R. Steinhauser, Systematic control design by optimizing a vector performance index, IFAC Proc. Vol. 12 (7) (1979) 113–117, http://dx.doi.org/10.1016/S1474-6670(17)65584-8, URL http://www.sciencedirect.com/science/article/pii/S1474667017655848.

[33] M. Jansen, G. Lombaert, M. Schevenels, O. Sigmund, Topology optimization of fail-safe structures using a simplified local damage model, Struct. Multidiscip. Optim. 49 (4) (2013) 657–666, http://dx.doi.org/10.1007/s00158-013-1001-y, URL http://link.springer.com/article/10.1007/s00158-013-1001-y.

[34] T. Schmidt, B. Kriegesmann, R. Seifried, Robust topology optimization of static systems with unilateral frictional contact, in: 8th European Congress on Computational Methods in Applied Sciences and Engineering, CIMNE, 2022, http://dx.doi.org/10.23967/eccomas.2022.237, URL https://www.scipedia.com/public/Schmidt_et_al_2022a.

[35] N. Strömberg, A. Klarbring, Topology optimization of structures in unilateral contact, Struct. Multidiscip. Optim. 41 (1) (2010) 57–64, http://dx.doi.org/10.1007/s00158-009-0407-z.

[36] A. Signorini, Questioni di elasticità non linearizzata e semilinearizzata, Rend. Mat. Appl. (7) 5 (18) (1959) 95–139.

[37] F. Ferrari, B.S. Lazarov, O. Sigmund, Eigenvalue topology optimization via efficient multilevel solution of the frequency response, Internat. J. Numer. Methods Engrg. 115 (7) (2018) 872–892, http://dx.doi.org/10.1002/nme.5829, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5829.

[38] A.J. Torii, J.R.d. Faria, Structural optimization considering smallest magnitude eigenvalues: a smooth approximation, J. Braz. Soc. Mech. Sci. Eng. 39 (5) (2017) 1745–1754, http://dx.doi.org/10.1007/s40430-016-0583-x.

[39] R.L. Fox, M.P. Kapoor, Rates of change of eigenvalues and eigenvectors, AIAA J. 6 (12) (1968) 2426–2429, http://dx.doi.org/10.2514/3.5008, Publisher: American Institute of Aeronautics and Astronautics _eprint.

[40] R.B. Nelson, Simplified calculation of eigenvector derivatives, AIAA J. 14 (9) (1976) 1201–1205, http://dx.doi.org/10.2514/3.7211, Publisher: American Institute of Aeronautics and Astronautics _eprint.

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

It is known that the previously discussed robust design optimization using the first-order second-moment method fails for several examples, meaning that no improvement of robustness compared to the deterministic result is obtained. This happens especially for problems, where the objective function is at a local minimum w.r.t. the random variables at the mean value, since then, the robust optimization is equal to the deterministic problem [114]. In practice, this problem usually occurs for random load orientations, because the optimization tool aligns the structure with the load (if possible) such that any variation of the load always leads to worse results.

Similar effects occur for the problem of random geometry. The structural optimization aims at finding the optimal geometry under given circumstances meaning that the gradient considering constraints is zero. In consequence, the gradient w.r.t. geometrical imperfections (probably using a different parameterization) might also be zero, if the corresponding variable is not restricted by a constraint. Often, only volume constraints are active, which do not restrict volume preserving changes of the geometry, such as bending-like deformations (see figure 4.1). In consequence, the gradient of the objective with respect to bending-like geometric imperfections is zero. However, buckling effects are often triggered by bending like imperfections. Hence, the objective function might have a local minimum but significant higher order terms with respect to geometric imperfections when using geometrically nonlinear analysis.

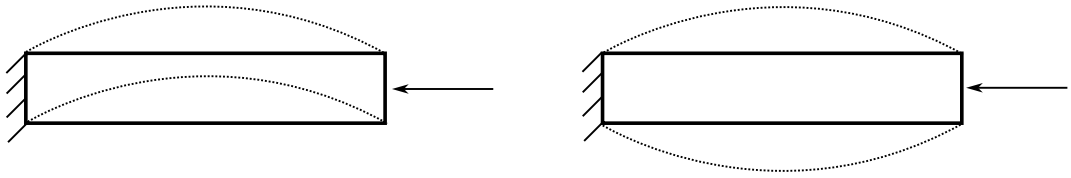


Figure 4.1: Volume preserving (left) and volume changing (right) geometric imperfections of an arbitrary structure. Under compression loads, the volume preserving imperfections are more prone to buckling.

To tackle the previously mentioned problems, the more accurate second-order fourth-moment method can be used [122]. The method is again derived as outlined in 3.4.2, but now using a second-order Taylor series. In difference to FOSM, the stochastic moments of the second-order fourth-moment method depend on second-order derivatives of the design response. In consequence, the gradient even depends on third-order derivatives leading to huge computational cost. In the following publication, [148], a more efficient and non-intrusive formulation similar to the principal sensitivity first-order second-moment method is derived and tested for several 3D examples using geometric nonlinear analyses.

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

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

Published online: 10 July 2024

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

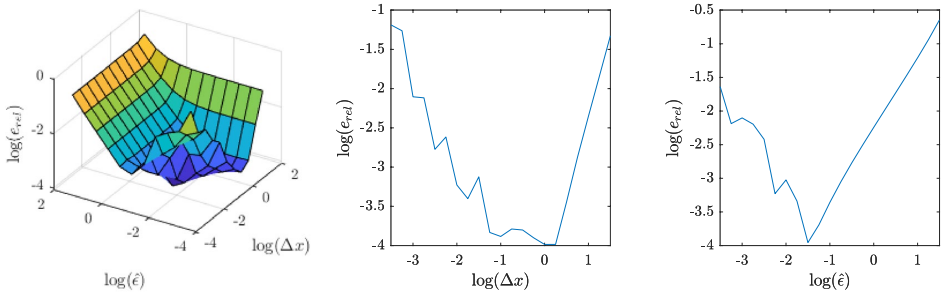
is used, where $e(\boldsymbol{X})$ denotes the error vector of the vector $\boldsymbol{X}$.

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

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

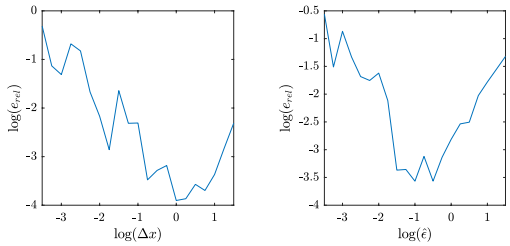
Using this normalization, $\hat{\epsilon}$ has no unit. In theory, it is possible to use different step sizes Δx_i and $\hat{\epsilon}_i$ for every i. However, from a practical point of view, it is hard to optimize many step sizes at the same time. Instead, all step sizes of the same type are chosen 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

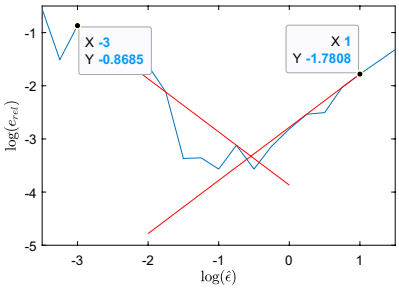


Fig. 3 Step size procedure: optimal step size is found at the intersection of two error lines

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, $\frac{d^3 f}{dx_i dy ds_i}$ 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:

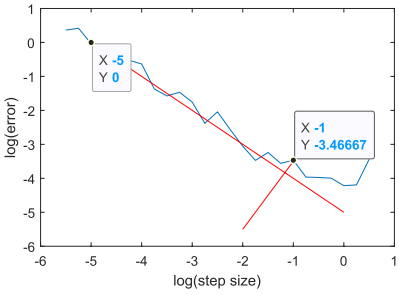


Fig. 4 Possible mistake when using the step size procedure: The initial guess is far away from the optimal step size. In this case, the estimated optimal step near one of the evaluation points

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

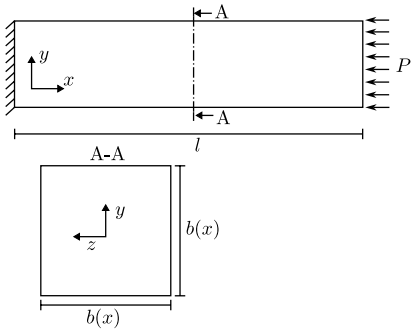


Fig. 5 Model and parameters of the compression bar

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

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

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

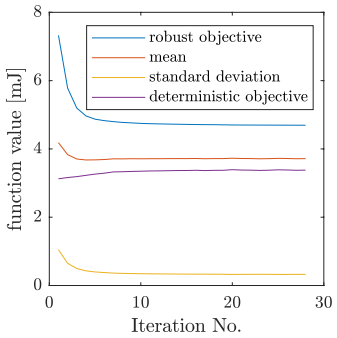


Fig. 7 History of the robust shape optimization of the compression bar

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

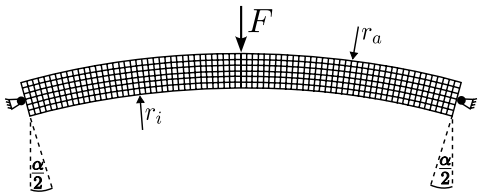
is expected that this structure is not optimal when considering uncertainties.

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.

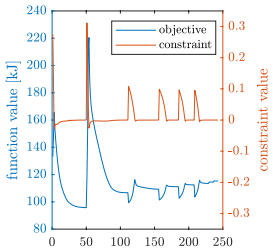
Mesh independency is achieved using the PDE filter given in Lazarov and Sigmund (2011). In order to get black and white designs, a regularized Heaviside projection given by

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

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 $s_{incr} = 1.1$.

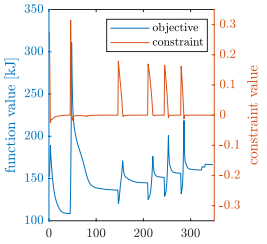
(a) Optimized design.



(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 $\tilde{f} = \mu_f + \kappa\sigma_f$ 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

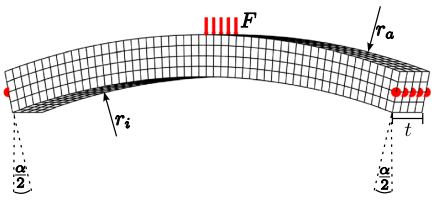
deterministic optimization. This is expected because 6 random parameters lead to $6 \cdot 4 + 1 = 25$ function evaluations per iteration. Parallel computation does not lead to significant improvements due to thermal limits of the laptop.

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

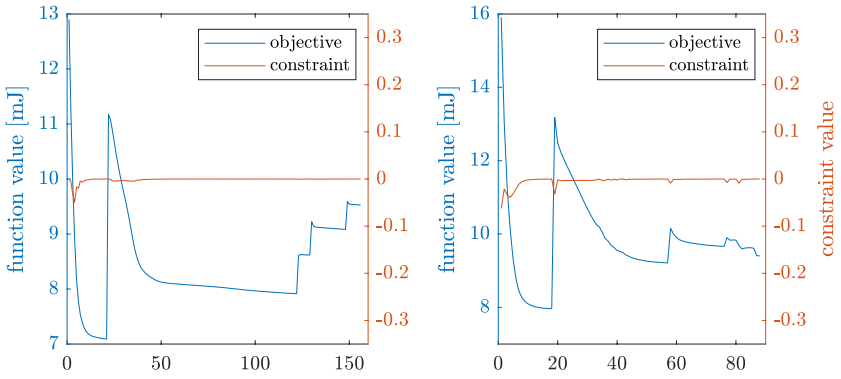


Fig. 15 Optimization history of the 3D shallow arc example. Left: deterministic optimized, right: robust optimized

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 $\tilde{f} = \mu_f + \kappa\sigma_f$ 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 latissimo 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

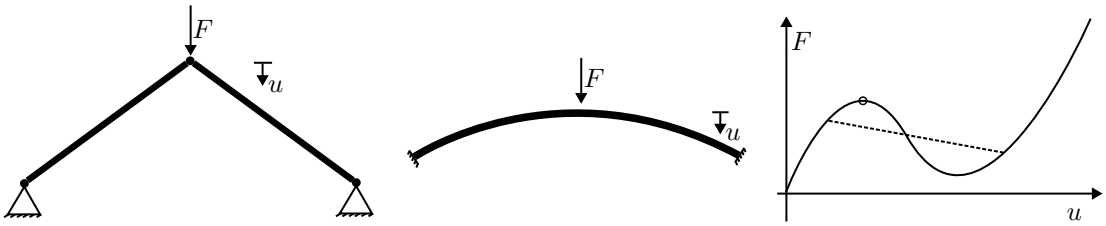


Figure 4.2: Problems where classic Taylor series-based approaches fail: snap-through load of perfect von-Mises truss (left) and shallow arch (middle). Right: An exemplary load displacement curve for perfect (solid) and imperfect (dashed) geometry and the maximum load (circle) of the perfect structure.

Figure 4.3: Design response of a deterministically optimized shallow arch with respect to the amplitude of a predefined imperfection (left) and a vizualization of the corresponding imperfection (middle). The grey structure represents a simple perfect geometry, the red structure represents the corresponding imperfect geometry at an imperfection amplitude of 10. The right figure shows the histogram of a Monte Carlo simulation considering multiple imperfection modes.

4.3 Multimodal first-order second-moment method

For many problems, the design response is accurately approximated by a second-order Taylor series, leading to accurate results when using the second-order fourth-moment method. However, it fails at the optimization for maximum load of snap-through problems, such as the von Mises truss [79] or the later considered shallow arc example depicted in figure 4.2. For an optimized design, the smooth load displacement curve depicted in figure 4.2 becomes non-differentiable under imperfections. The transition between smooth and non-differentiable load displacement curves suddenly occurs if the geometric imperfection exhibits a specific threshold value. In consequence, the response function of the maximum load with respect to an imperfection amplitude of an asymmetric geometric imperfection (both depicted in figure 4.3) becomes discontinuous with a plateau at zero imperfection. A corresponding histogram depicted in figure 4.3 shows bimodal behavior. Similar observations are also done for the optimization of stiffened panels under a similar objective function [117]. In consequence, any approach based on a Taylor series approximation fails for these kinds of problems.

To address the problem of discontinuous response functions leading to multimodal probability distributions, a new framework called "multimodal first-order second-moment method" is proposed and has been preliminary submitted for publication in [149]. There, the author of this thesis contributed critical aspects of the methodology and provided the shallow arch example, which is presented in the following.

4.3.1 Basic framework

The multimodal first-order second-moment method is based on the idea to approximate the true objective function $f(\boldsymbol{y}, \boldsymbol{x})$ by a piecewise (first-order) Taylor approximation

$$f(\boldsymbol{y}, \boldsymbol{x}) \approx \hat{f}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i f_i(\boldsymbol{y}, \boldsymbol{x}) \tag{4.1}$$

with

$$f_i(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) = f(\boldsymbol{y}, \boldsymbol{x}_i) + \nabla f(\boldsymbol{y}, \boldsymbol{x}_i)^T \cdot (\boldsymbol{x} - \boldsymbol{x}_i) + \dots & : \boldsymbol{x} \in \Omega_i \\ 0 & : \boldsymbol{x} \notin \Omega_i \end{cases} \tag{4.2}$$

and the center $\boldsymbol{x}_i$ of the Taylor series T_f of the function f in the domain Ω_i. Like for the classic first-order second-moment method, the stochastic moments are obtained by inserting the approximated function (4.1) into the analytic equations for mean (3.1) and variance (3.2). However, the resulting integrals are solved piecewise, such that the mean reads

$$\mu_{\hat{f}} = \sum_i \int_{\Omega_i} T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) p(\boldsymbol{x}) d\boldsymbol{x} \tag{4.3}$$

and the variance reads

$$\sigma^2_{\hat{f}} = \sum_i \int_{\Omega_i} \left(T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) - \mu_{\hat{f}} \right)^2 p(\boldsymbol{x}) d\boldsymbol{x} \quad . \tag{4.4}$$

Defining the sectionwise probability density function $p_i(\boldsymbol{x})$ of section Ω_i as

$$p_i(\boldsymbol{x}) = \begin{cases} p(\boldsymbol{x}) \cdot \frac{1}{w_i} & : \boldsymbol{x} \in \Omega_i \\ 0 & : \boldsymbol{x} \notin \Omega_i \end{cases} \tag{4.5}$$

with $w_i = \int_{\Omega_i} p(\boldsymbol{x}) d\boldsymbol{x}$ being the probability of a sample being in the domain Ω_i leads to

$$\mu_{\hat{f}} = \sum_i w_i \int_{\Omega_i} T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) p_i(\boldsymbol{x}) d\boldsymbol{x} = \sum_i w_i \mu_{\hat{f},i} \quad . \tag{4.6}$$

The value $\mu_{\hat{f},i}$ represents the mean of the approximated function value considering only the part of the probability distribution in section Ω_i. Using these definitions, the variance reads

$$\begin{aligned} \sigma^2_{\hat{f}} &= \sum_i w_i \int_{\Omega_i} \left(T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) - \mu_{\hat{f},i} + \mu_{\hat{f},i} - \mu_{\hat{f}} \right)^2 p_i(\boldsymbol{x}) d\boldsymbol{x} \\ &= -\mu^2_{\hat{f}} + \sum_i w_i \left(\int_{\Omega_i} \left(T_f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_i) - \mu_{\hat{f},i} \right)^2 p_i(\boldsymbol{x}) d\boldsymbol{x} + \mu^2_{\hat{f},i} \right) \quad , \\ &= -\mu^2_{\hat{f}} + \sum_i w_i \left(\sigma^2_{\hat{f},i} + \mu^2_{\hat{f},i} \right) \end{aligned} \tag{4.7}$$

where $\sigma^2_{\hat{f},i}$ represents the variance of the approximated function value considering only the part of the probability distribution in section Ω_i.

For solving the equations (4.6) and (4.7), the section-wise mean values and variances are computed using the first-order second-moment method by

$$\mu_f \approx \mu_{\hat{f}} = \sum_i w_i f(\boldsymbol{y}, \mu_{\boldsymbol{x},i}) \tag{4.8}$$

and

$$\sigma_f^2 \approx \sigma_{\hat{f}}^2 = \sum_i w_i \left(\nabla^T f(\boldsymbol{y}, \mu_{\boldsymbol{x},i}) \boldsymbol{\Sigma}_{\boldsymbol{x},i} \nabla f(\boldsymbol{y}, \mu_{\boldsymbol{x},i}) + f(\boldsymbol{y}, \mu_{\boldsymbol{x},i})^2 \right) - \mu_{\hat{f}} \quad , \tag{4.9}$$

where $\mu_{\boldsymbol{x},i}$ is the mean of the domain-wise random distribution and $\boldsymbol{\Sigma}_{\boldsymbol{x},i}$ the covariance matrix of the domain-wise random distribution of domain Ω_i. As far as the domain definition does not change during the optimization, these values are computed once in a preprocessing step. The general procedure may be extended to higher order approaches respectively. In this case, the section-wise values are computed, e.g., using the second-order fourth-moment method.

4.3.2 Domain definition and computation of domain-wise stochastic moments

The results obtained using the multimodal first-order second-moment method highly depend on the chosen domain, leading to different mean values $\mu_{\boldsymbol{x},i}$ and covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{x},i}$ of the subdomains Ω_i. If the general shape of the true design response is known, the domains can be chosen particularly for this problem. However, in general this knowledge is not available. Additionally, the domains possibly change during the optimization process. However, in many cases some engineering knowledge is available. For example it is known that buckling is usually triggered by asymmetric imperfections in the corresponding bars. This knowledge may be used to generate the domains. To keep the framework as simple as possible, it is proposed to generate the domains by splitting the overall domain along the axes or the bisectors of (transformed) uncorrelated random variables, as depicted for a 2-variate problem in figure 4.4. The corresponding stochastic moments of the random variables are found by Monte Carlo simulations or can be computed analytically for specific random distributions.

Split along axes

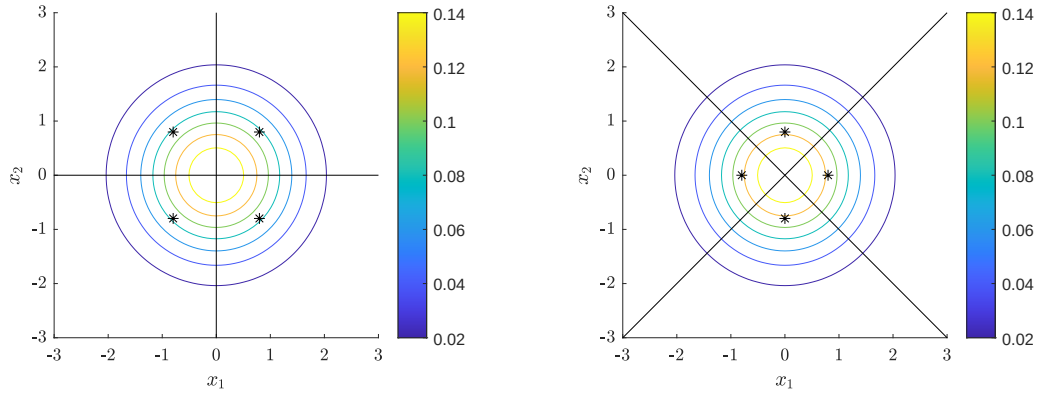
If the split is done along the axes as depicted in figure 4.4a, the mean of the random variables $\boldsymbol{x}$ in the domains Ω_i are computed for the simple two-variate case by

$$\mu_{\boldsymbol{x},1} = \int_0^\infty \int_0^\infty \boldsymbol{x} p(x_1, x_2) dx_1 dx_2 \cdot \frac{1}{\int_0^\infty \int_0^\infty p(x_1, x_2) dx_1 dx_2} \tag{4.10}$$

$$\mu_{\boldsymbol{x},2} = \int_0^\infty \int_{-\infty}^0 \boldsymbol{x} p(x_1, x_2) dx_1 dx_2 \cdot \frac{1}{\int_0^\infty \int_{-\infty}^0 p(x_1, x_2) dx_1 dx_2} \tag{4.11}$$

$$\mu_{\boldsymbol{x},3} = \int_{-\infty}^0 \int_{-\infty}^0 \boldsymbol{x} p(x_1, x_2) dx_1 dx_2 \cdot \frac{1}{\int_{-\infty}^0 \int_{-\infty}^0 p(x_1, x_2) dx_1 dx_2} \tag{4.12}$$

$$\mu_{\boldsymbol{x},4} = \int_{-\infty}^0 \int_0^\infty \boldsymbol{x} p(x_1, x_2) dx_1 dx_2 \cdot \frac{1}{\int_{-\infty}^0 \int_0^\infty p(x_1, x_2) dx_1 dx_2} \quad . \tag{4.13}$$



(a) Split along axes

(b) Split along bisectors.

Figure 4.4: Proposed splits for a normal distribution in 2D. The markers indicate the mean of the domain random distribution while the isolines represent the underlying probability density function.

For the more general multivariate case, additional dimensions with integral borders $[0, \infty]$ or $[-\infty, 0]$ are added. If no split is done in a specific dimension, the integral borders for the corresponding dimension stay at $[-\infty, \infty]$. Assuming the random variables x_1, x_2 are independent from each other (i.e., uncorrelated for a normal distribution), the equation for the first domain reads

$$\mu_{\boldsymbol{x},1} = \begin{pmatrix} \frac{\int_0^\infty x_1 p(x_1) dx_1}{\int_0^\infty p(x_1) dx_1} & \frac{\int_0^\infty x_2 p(x_2) dx_2}{\int_0^\infty p(x_2) dx_2} \end{pmatrix}^T \quad . \tag{4.14}$$

Considering a standard normal distribution, the solution is found as $\mu_{\boldsymbol{x},1} = \begin{pmatrix} \sqrt{\frac{2}{\pi}} & \sqrt{\frac{2}{\pi}} \end{pmatrix}^T$. More generally, the mean of a domain Ω_i in the normal-distributed multivariate case is

$$\mu_{\boldsymbol{x},i} = \begin{pmatrix} \mu_{x_1,i} & \mu_{x_2,i} & \dots \end{pmatrix}^T \tag{4.15}$$

with

$$\mu_{x_j,i} = \begin{cases} \sqrt{\frac{2}{\pi}} & x_j > 0 \forall \boldsymbol{x} \in \Omega_i \\ -\sqrt{\frac{2}{\pi}} & x_j < 0 \forall \boldsymbol{x} \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

which means that $\mu_{\boldsymbol{x},i,j} = 0$, if no split is done in dimension j, $\mu_{\boldsymbol{x},i,j} = \sqrt{\frac{2}{\pi}}$ for the positive domain regarding dimension j and $\mu_{\boldsymbol{x},i,j} = -\sqrt{\frac{2}{\pi}}$ for the negative domain regarding dimension j.

Similarly, the domain covariance matrix is found by integrating (3.6) for the subdomain Ω_i leading to

$$\boldsymbol{\Sigma}_{\boldsymbol{x},i} = \begin{pmatrix} 1 - \frac{2}{\pi} & 0 & 0 & \dots \\ 0 & 1 - \frac{2}{\pi} & 0 & \dots \\ 0 & 0 & 1 - \frac{2}{\pi} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{4.17}$$

for the simple case of standard normal distributed random variables. In the more general case of arbitrarily distributed random variables, the variables can be transformed to correlated normal distributed variables using the Rosenblatt transformation [150] and further transformed to standard normal distributed random variables using the previously described Karhunen-Loéve expansion or the EOLE method.

Defining the sub-domains Ω_i by splitting the whole domain using splits along the axes, the number of domains doubles per split leading to an exponential growth of required domain-wise probabilistic analyses. Due to the exponential growth, the computational cost becomes infeasible already for a moderate number of random variables (e.g., 10 variables lead to 1000 domains).

Split along bisectors

To reduce the required number of domains, the splits may be done along bisectors as depicted in figure 4.4b, such that only two subdomains are required per variable. Here, the domain-wise stochastic moments are computed similarly as in the case of splits along the axes, however the integral bounds are changed leading to

$$\mu_{\boldsymbol{x},1} = \frac{\int_0^\infty \int_{-x_1}^{x_1} \cdots \int_{-x_1}^{x_1} \boldsymbol{x} p(x_1, x_2, \ldots, x_N) dx_N \ldots dx_2 dx_1}{\int_0^\infty \int_{-x_1}^{x_1} \cdots \int_{-x_1}^{x_1} p(x_1, x_2, \ldots, x_N) dx_N \ldots dx_2 dx_1} \tag{4.18}$$

for the mean of the first domain. The mean values for the other domains are computed accordingly. Under the assumption of standard normal distributed random variables, the integral for the first domain simplifies to

$$\mu_{\boldsymbol{x},1} = \begin{pmatrix} 2N \int_0^\infty x_1 p(x_1) \cdot (2\Phi(x_1) - 1)^{N-1} dx_1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} , \tag{4.19}$$

where N represents the number of splits and $\Phi(x_1)$ is the cumulative random density function. For a more detailed derivation, the reader is referred to [149]. The covariance matrix is computed in a similar way. For the case of standard normal distributed random variables, there is no correlation between different variables and the variances $\sigma^2_{x_j,1}$ of domain Ω_1 read

$$\sigma^2_{x_j,1} = \begin{cases} 2N \int_0^\infty (x_1 - \mu_{x_1,1})^2 \cdot p(x_1) \cdot (2\Phi(x_1) - 1)^{N-1} dx_1 & j = 1 \\ 2N \int_0^\infty (2\Phi(x_1) - 2x_1 p(x_1) - 1) \cdot p(x_1) \cdot (2\Phi(x_1) - 1)^{N-2} dx_1 & j \neq 1 \end{cases} . \tag{4.20}$$

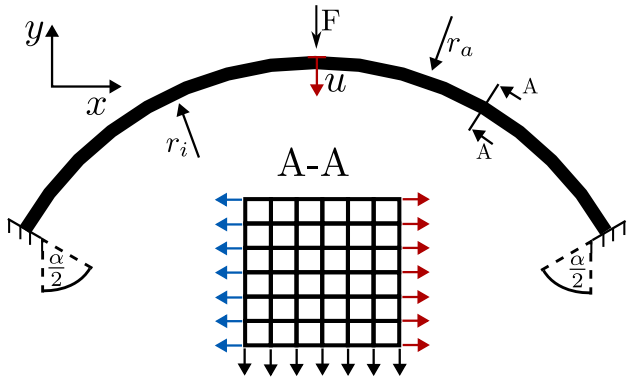
In practice, the occurring integrals are solved numerically.

4.3.3 Gradient computation

For a robust design optimization, the gradient of the stochastic moments of the response function (4.8) and (4.9) are required. Using direct differentiation, the gradients read

$$\frac{d\mu_{\hat{f}}}{d\boldsymbol{y}} = \sum_i w_i \nabla f(\mu_{\boldsymbol{x},i}) \tag{4.21}$$

$$\frac{d\sigma^2_{\hat{f}}}{d\boldsymbol{y}} = -\frac{d\mu_{\hat{f}}}{d\boldsymbol{y}} + \sum_i w_i \nabla f(\mu_{\boldsymbol{x},i}) + \frac{d\sigma^2_{\hat{f},i}}{d\boldsymbol{y}} \tag{4.22}$$



Inner radius r_i	180 mm
Outer radius r_a	186 mm
thickness t	6 mm
Angle α	50°
Young's modulus	2 GPa
Poisson ratio	0.3
displacement steps u_i	$\{2 : 0.075 : 8\}$ mm
correlation length l_c	80 mm
standard deviation σ	0.5 mm
filter radius axial	9 mm
filter radius radial	3 mm
filter radius z-direction	3 mm

Figure 4.5: Sketch of the model and parameters of the shallow arc example. The nodal coordinates of the lower (black) and the side nodes (blue, red) are moved into normal direction. The three sets are filtered independently of each other.

To compute the gradient of the domain-wise variance $\sigma^2_{\hat{f},i}$, all methods for the classic first-order second-moment method discussed in section 4.1 may be used.

4.3.4 Application to shallow arc example

The presented multimodal first-order second-moment method is applied to the shape optimization of a shallow arc with respect to the maximum load of the snap-through problem under volume constraints. As mentioned before, designs generated by this optimization problem might have a discontinuous response function with respect to asymmetric-imperfections caused by buckling effects (see figure 4.3).

A sketch of the model, the shape optimization parameterization and all model parameters are collected in figure 4.5. The problem consists of a shallow arc with the inner radius $r_i = 180mm$, the outer radius $r_a = 186mm$, the thickness $t = 6mm$ and the opening angle $\alpha = 50°$. It is loaded by a line load F, distributed over the top center line and fully clamped at the sides. The model is discretized by a body-fitted mesh using $320 \times 12 \times 12$ fully integrated bi-linear hexahedral finite elements. The finite element simulation is based on a geometrically nonlinear analysis, using the Green-Lagrange strain measure in combination with the Saint Venant-Kirchhoff constitutive law. The nonlinear problem is solved using a displacement-driven solver [151], where the displacement u at the center node of the line load F is applied in 80 equidistant steps until the final displacement $u_{max} = 8mm$ is reached. The linear subproblems are solved using the multigrid-preconditioned conjugate gradient method [60]. As material, standard 3D printing resin with the Young's modulus $E = 2$GPa and the Poisson's ratio $\nu = 0.3$ is considered.

The optimization problem is parameterized using perturbations of the nodes as design variables, such that the coordinates $\boldsymbol{z}_i$ of node i reads

$$\boldsymbol{z}_i = \boldsymbol{z}_{i,0} + \Delta\boldsymbol{z}_i \quad . \tag{4.23}$$

As discussed in section 2.3, movements of inner nodes as well as tangential movements of surface nodes do not change the geometry of the model but lead to numerical errors. Therefore, only

normal perturbations of the surface nodes are used as design variables and the inner nodes are updated using Laplacian smoothing [42]. As depicted in figure 4.5, only the nodal coordinates of the lower surface, the front surface ($z = 3$mm) and the back surface ($z = -3$mm) are considered in the optimization, whereas the top nodes and the clamped nodes are locked. This choice is made for several reasons: First, the possibility of changing the top nodes and the clamped nodes would also move the loads and boundary conditions, which is usually not accepted. Additionally, the optimizer would focus on changing the opening angle α instead of improving the shape. Finally, the restriction of the upper surface may be interpreted as a boundary condition. The top surface might not be changed because it has some other functions, e.g., it represents the running way of a bridge.

To maintain mesh quality, the design variables are filtered using the vertex morphing approach with a first-order filter function and first-order displacement fields [44]. However, the three sets of nodes are filtered independently of each other, since they represent different shape features. Due to the high aspect ratio of the model, an anisotropic filter function is used, where the filter radius is set to 9mm in the tangential direction and to 3mm in the other directions (radial and out-of-plane). Additionally, the smoothing operator is modified, such that smoothing does not lead to x- or y-movements of the surface nodes. Considering these aspects, the shape update reads

$$\boldsymbol{z}_i = \boldsymbol{z}_{i,0} + \text{smooth}(\text{filter}(\boldsymbol{y})) \tag{4.24}$$

with the design variables $\boldsymbol{y}$, the filter function "filter" and the Laplacian smoothing function "smooth". The parameter boundaries are chosen such that the thickness of the structure locally increases or decreases by up to 60% of the original thickness.

For the initial design (i.e. $\Delta \boldsymbol{z}_i = 0 \forall i$), the load displacement curve depicted in figure 4.6 is obtained. It shows a linear increase for small displacements whereas it becomes nonlinear for growing displacements. At $u = 7.5$mm, the maximum is reached and afterwards, the function decreases to its local minimum. The goal of the optimization is to increase the load maximum, since it defines, how much load the structure can carry before stability failure occurs. However, the maximum function is not differentiable and hence, it is not suitable for a gradient-based optimization. To circumvent this challenge, the aggregation approach is chosen. Therefore, the forces F at the displacements $u_i \in \{2 : 0.075 : 8\}$mm (marked with red dots in figure 4.6) are aggregated by the p-norm function

$$f = \frac{1}{n} \left(\sum_{i=1}^{n} F(u_i)^p \right)^{\frac{1}{p}} \approx \max_u (F(u)) \tag{4.25}$$

with p chosen as $p = 16$. The relatively low values of u are chosen because bifurcation buckling might lead to a significant shift of the maximal load, such that high values are already in the post-buckling regime. The computation of the force $F(u_i)$ and its gradient are explained in the appendix A.

Using these definitions, the overall deterministic optimization problem reads

$$\begin{aligned} &\min_{\boldsymbol{y}} -f(\boldsymbol{y})) \\ &\text{s.t.} \\ &-1.8\text{mm} \leq y_i \leq 1.8\text{mm} \quad \forall i \quad , \\ &\frac{V(\boldsymbol{y}) - V_0}{V_0} \leq \mathbf{0} \end{aligned} \tag{4.26}$$

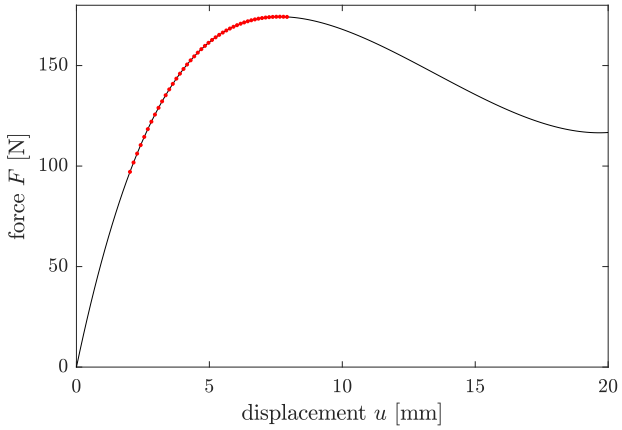


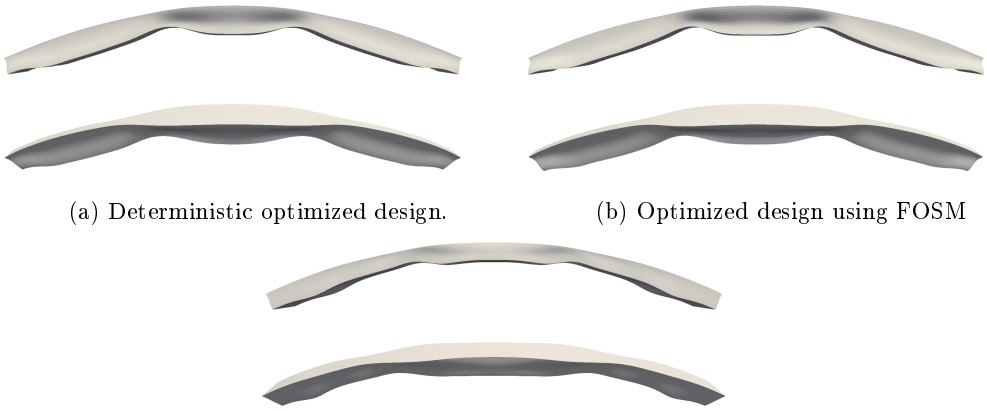
Figure 4.6: Load-displacement curve of the initial design. The loads marked with red dots are aggregated using a p-norm for the objective function.

where the function $V(\boldsymbol{y})$ represents the design-dependent volume and V_0 represents the initial volume.

For the robust optimization, geometric imperfections are modeled by considering uncertain nodal coordinates. In difference to the design variables, no nodes or degrees of freedom are excluded, since also surfaces that are not optimized might still be influenced by imperfections. Correlation is modeled using a normal distributed square exponential random field with the correlation length $l_c = 80\text{mm}$ and the standard deviation $\sigma = 0.5\text{mm}$. It is assumed that the different degrees of freedom ($x-, y-, z-$ direction) are uncorrelated. To transform the correlated random variables to standard normal distributed random variables, the random field is discretized using the EOLE method. To reduce the memory demands for the covariance matrix, only the first 15 components of the EOLE discretization are considered, reducing the memory demands from 200GB to 20MB at the cost of less than 0.4% error of the variance. For the evaluation using the multimodal first-order second-moment method, the splits are done along the bisectors, since using the splits along axes would require $2^{15} \approx 32'000$ FOSM evaluations per multimodal evaluation leading to infeasible computational cost. Using the split along bisectors, only 30 FOSM evaluations are required per solve, which can be perfectly parallelized to 30 workers. Finally, the robust design optimization formulation reads

$$\begin{aligned} &\min_{\boldsymbol{\rho}} -\mu_f(\boldsymbol{y})) + 3\sigma_f(\boldsymbol{y})) \\ &\text{s.t.} \\ &-1.8\text{mm} \leq y_i \leq 1.8\text{mm} \quad \forall i \quad \cdot \\ &\frac{V(\boldsymbol{y}) - V_0}{V_0} \leq \mathbf{0} \end{aligned} \tag{4.27}$$

The structure is optimized in a deterministic optimization, a robust optimization using the classic first-order second-moment method and the multimodal first-order second-moment method. For the FOSM approximations, the gradients are computed using the principal sensitivity approach



(a) Deterministic optimized design.

(b) Optimized design using FOSM

(c) Optimized design using multimodal FOSM.

Figure 4.7: Optimized designs of the shallow arch. The results are shown from the front (top figure) and slightly tilted around the x-axis to show the 3D-structure. The figure is taken from [149].

presented in [145]. All optimizations are done using the globally convergent method of moving asymptotes [13] with the initial asymptote value $s_{ini} = 0.05$. The optimization is stopped, if the relative function value change between two subsequent iterations is less than 0.01% and all constraints are fulfilled, leading to the designs depicted in figure 4.7 with the resulting values printed in table 4.1. There, all designs are evaluated deterministically, using the first-order second-moment method, the multimodal first-order second-moment method and the Monte Carlo method with a sample size of 1500.

		probabilistic evaluation approach					
		FOSM		MM-FOSM		Monte Carlo	
optimization approach	f	μ_f	σ_f	μ_f	σ_f	μ_f	σ_f
determ. optimization	395 N	395 N	19 N	288 N	94 N	140 N	30 N
RDO / FOSM	415 N	415 N	20 N	290 N	98 N	141 N	30 N
RDO / MM-FOSM	169 N	169 N	6.8 N	167 N	7.2 N	164 N	6.2 N

Table 4.1: Numeric results of the optimized shallow arc evaluated deterministically, using the first-order second-moment method (FOSM), the multimodal first-order second-moment method (MM-FOSM) and the Monte Carlo method with a sample size of 1500.

Overall, the deterministic optimization and the robust optimization using the first-order second-moment method lead to the same design, whereas the multimodal first-order second-moment method leads to a different design. This is explained by the fact, that the design response has a discontinuous shape like shown in figure 4.3, where the classic first-order second-moment method does not detect the drop of the function value. In difference, the multimodal approach is tailored to this specific kind of problem, although the exact discontinuity is not known. If at least one evaluation point triggers buckling of the structure, the optimizer detects a strongly increased standard deviation and worse mean. Hence, the focus is set on improving the worst evaluation point one side while

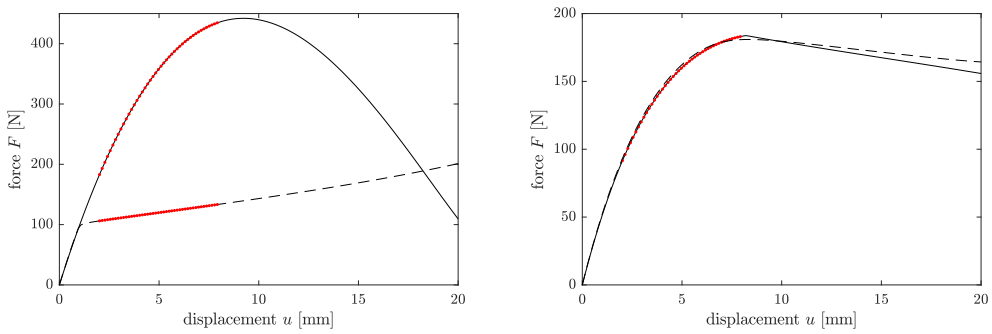


Figure 4.8: Load-displacement curve of the deterministic optimized design (left) and of the robust optimized design using the multimodal first-order second-moment method (right) without (solid) and with imperfections (dashed). The loads marked with red dots are aggregated using a p-norm for the objective function.

reducing the differences between the evaluation points (i.e. the variance), leading to a different design.

The deterministic (and FOSM) design is a symmetric model based on three parts connected by hinges, which are barely able to transfer moments. However, in the ideal configuration, no moments occur at the hinges due to symmetry. The middle part is similar to an I-shaped profile which is well suited for carrying the occurring bending loads. Due to the hinge mechanisms, mainly normal forces are transferred to the side arms, which are thickened on the bottom. Due to this thickening, the normal forces are transferred at a high angle to the lower part of support. From engineering knowledge it can be expected that this design performs well under ideal conditions. However, significant (torsional or bending) moments occur in presence of asymmetric imperfections, which are barely transferred by the weak joints, making the structure prone to asymmetric buckling. In consequence, the expected value obtained with the Monte Carlo method is 70% lower than the deterministic function value.

In difference, the design optimized using the multimodal first-order second-moment method has a different structure. Again, a nearly symmetric design is found, which consists of three parts. However, the joints between the parts have a slightly higher bending stiffness and the middle part is much longer. Due to the design, the side parts always also carry bending moments. In consequence, these parts are constructed accordingly, leading to much worse performance in the deterministic case. However, the robust optimized structure does not suffer from the assumed imperfections. In fact, the mean is similar to the deterministic value and a low standard deviation is obtained.

The differences between both results are also seen in the final load displacement curves depicted in figure 4.8. Ideally, the deterministic result shows a non-buckling shape, which drastically changes in the presence of imperfections. In difference, the robust design already shows buckling behavior in the deterministic case. In consequence, imperfections only change the buckling shape, but do not harm the overall performance. Besides, note that the optimization moved the maximum load to displacements that are higher than the considered displacements in the objective function (4.25). This especially holds for the imperfect designs. In consequence, finally, the maximum load is not directly optimized. However, also considering higher displacements is not expected to significantly change the results.

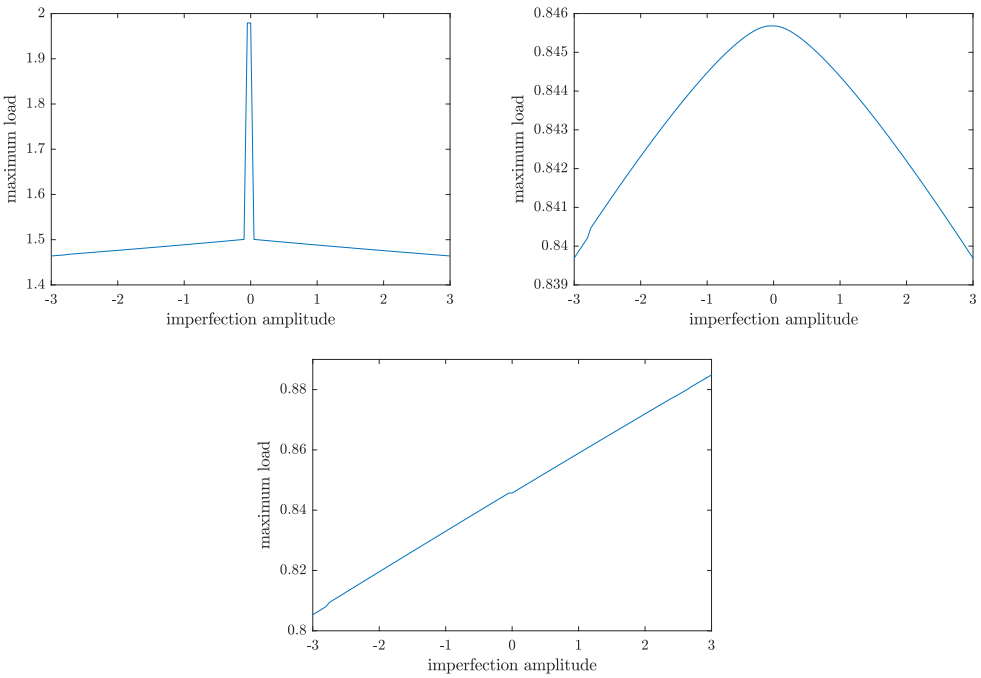


Figure 4.9: Response function with respect to the first EOLE mode of the deterministically optimized design (top left) and the robust optimized design using the multimodal approach (top right). The bottom figure shows the response with respect to the (volume changing) second EOLE mode for the robust optimized design.

Comparing the accuracy of the approximations of stochastic moments reveals that the multimodal approach is more accurate than the classic first-order second-moment method. Especially it is able to detect the effect of a discontinuous response function at the deterministically optimized design, although the moment approximations are still very inaccurate. It is expected that these inaccuracies are caused by badly chosen splits. In fact, more detailed investigations reveal that choosing the splits along the axes might lead to more accurate results. In difference to the deterministically optimized design, all approaches are relatively accurate for the successfully robust optimized design. In the robust optimization, the standard deviation is penalized, which means that jumps in the response function lead to high objective values. As depicted in figure 4.9, the discontinuous jumps are reduced for the first EOLE component, while still nonlinearities exist. However, these nonlinear components are much less dominant compared to linear components that occur for volume-changing imperfection modes (here mode 2). Hence, also the classic first-order second-moment method performs well.

Chapter 5

Robust design optimization using experimental data

As the previous sections focused on the development of scalable approaches, the used examples have either been taken from literature or were highly constructed to show different aspects of the derived methods. Especially, the considered uncertainties were not based on empirical data.

In the following, the derived methods are applied to a more realistic example of a wooden three-point bending beam, where the dimensions, loads and boundary conditions are tailored to a physical experimental setup. Spatially uncertain material properties (i.e., Young's modulus and material orientation) are measured experimentally and the final design is fabricated using a laser cutting machine. The author chose spruce wood as base material for several reasons: First, it is possible to produce a large amount of test specimens at a very low cost (0.1 euros for material test, 5 euros for optimized design). Second, wood as a natural material is expected to show much higher uncertainties than standard engineering materials, which makes a robust optimization more relevant. Finally, the author believes that wood is an underestimated high performance material, which should be considered more frequently for lightweight design.

5.1 On the application of the multigrid method to topology optimization with orthotropic material with varying orientation

First investigations on the application of robust topology optimization of wooden structures reveal challenges in the solution of the governing finite element equations. In general, topology optimization problems require a rather fine discretization, leading to infeasible computational costs when using direct solution methods to the finite element equations. Therefore, multigrid-preconditioned iterative solvers, such as the multigrid-preconditioned conjugate gradient method became the state of the art for such problems, leading to low solution times [60]. However, the author's publication in this chapter [152] shows that standard approaches fail for anisotropic materials, such as wood. To tackle this issue, the occurring problems are analyzed, different existing solutions are compared and a new smoothing method for the multigrid preconditioner is provided.

Structural and Multidisciplinary Optimization (2025) 68:158
https://doi.org/10.1007/s00158-025-04102-y

RESEARCH PAPER

On the application of the multigrid method to topology optimization with orthotropic material with varying orientation

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

Received: 4 March 2025 / Revised: 11 June 2025 / Accepted: 27 July 2025

Abstract
Multigrid methods are known to show good performance for most topology optimization problems, but suffer from anisotropies. The main problems are caused by poor performance of the smoothing algorithms. This paper analyzes different standard and advanced smoothing algorithms using the local Fourier analysis with regard to orthotropic materials in the linear elasticity problem. The results are generalized to real problems and contrast-rich topology designs using numeric experiments. Based on the analyses, recommendations are given and a new spatial Jacobi smoothing algorithm is presented. The results are finally approved by a topology optimization example and a material orientation and topology optimization example.

Keywords Multigrid method · Topology optimization · Material orientation and topology optimization · Orthotropic material · Large scale

1 Introduction

Topology optimization has become a useful and widely accepted approach for computational design. While the first applications of topology optimization were mainly restricted to 2D problems with less than 10,000 voxel finite elements (Buhl et al. 2000; Hammer and Olhoff 2000), increasing computational resources and new algorithms led to a wider applicability of this method. A huge effort has been taken for the solution of linear systems of equations arising from the finite element method, since this part is responsible for the main computation time and memory demands. In the context of parallel computing and the use of high-performance clusters, domain decomposition methods help to distribute the computational load to several workers. By doing this, also medium sized 3D-problems were solved in the early topology optimization days (Borrvall and Petersson 2001; Mahdavi et al. 2006; Evgrafov et al. 2008). Other approaches focused on improving iterative Krylov subspace solvers, for example, by approximate reanalysis techniques (Amir et al. 2009) and recycling of parts (Wang et al. 2007; Amir and Sigmund 2011).

A huge increase in efficiency for solving linear systems of equations was achieved by the application of the multigrid method. Although this method has been known for many decades (see, e.g., Shaidurov (1995)) and it was early applied to topology optimization (Dreyer et al. 2000), it gained attention after the first educational paper by Amir et al. (2014). There, a Galerkin projection-based geometric multigrid method is used as a preconditioner for the conjugate gradient method. With this, it is possible to run topology optimization problems that have been called "large scale" in the early days on a simple laptop. Several publications followed, applying the multigrid method to open-source code (Aage et al. 2015), to giga-scale problems (Aage et al. 2017) and to shell structures (Träff et al. 2021). Others present highly efficient implementations exploiting the narrow band structure of the stiffness matrix (Liu et al. 2018) or using graphic processor units (Wu et al. 2016; Herrero-Pérez and Martínez Castejón 2021; Padhi et al. 2023).

Despite these achievements, there are still open challenges. Besides the problem of high-contrast designs, anisotropic behavior of the partial differential equation also leads to significantly reduced convergence rates of the multigrid

Responsible editor: Marco Montemurro

✉ Jan Christoph Krüger
jan.krueger@tuhh.de

Benedikt Kriegesmann
benedikt.kriegesman@tuhh.de

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

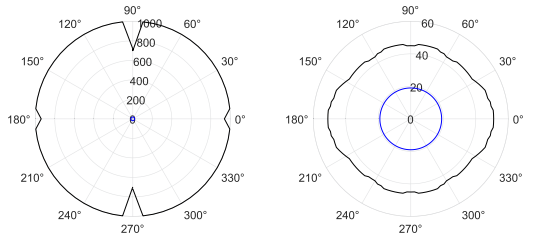
Published online: 21 August 2025

solution (Trottenberg et al. 2007). The (discretized) partial differential equation becomes anisotropic when using orthotropic materials or in the context of stretched finite elements. This effect usually does not occur in topology optimization since often the problem is solved on a voxel mesh with isotropic material (e.g., 3D printed steel). However, anisotropic problems also occur in reality.

Application of the simple multigrid approach presented in Amir et al. (2014) with the same settings (damping factor $\omega = 0.6$) to the same cantilever example but with carbon fiber-reinforced polymer (CFRP) as material leads to much worse performance. Figure 1a shows the numbers of required multigrid iterations for different fiber orientations at the initial design. The black curve corresponds to the CFRP problem, and the blue curve corresponds to the isotropic problem from Amir et al. (2014). For all orientations, the number of iterations sticks at approximately 1000, which is also the maximum allowed number of iterations.

To retrieve acceptable performance for all orientations, the damping factor must be significantly reduced to $\omega = 0.3$. As shown in Fig. 1b , this reduces the number of iterations to approximately 50. However, the iteration number is still significantly higher than the number of iterations required for isotropic materials.

For the case of stretched finite elements, Peetz and Elbanna (2021) show that using algebraic multigrid reduces the number of solver iterations. However, they conclude that the additional effort for setting up the algebraic operators is much higher than the gain in 3D. On the other hand, already early publications showed that plane-smoothing (Oosterlee 1997) or semi-coarsening (Morano et al. 1998) are very effective for a priori known constant anisotropies. However, the assumption of a priori known constant anisotropies is not valid in practice.



(a) CFRP, $\omega = 0.6$ (b) CFRP, $\omega = 0.3$

Fig. 1 Number of iterations to solve a linear equation using multigrid preconditioned conjugate gradients with Jacobi smoothing and a relative residual of less than 10^{-10}. The radius corresponds to the number of iterations, the angle represents the rotation angle of the material. The blue plot corresponds to an isotropic material with Poisson's ratio 0.3. The black plot corresponds to a carbon fiber reinforced polymer

For example, orthotropic mechanical properties occur in natural organic material such as wood. A topology optimization of wooden structures is, for example, presented in Ma et al. (2021) and de Vito et al. (2023). In general, the orientation is not constant over the domain since effects of branches and unsteady growth lead to spatially varying fiber orientations. Additionally, the two weak directions are oriented into the radial and tangential directions of the trunk (Thibaut et al. 2001; Da Silva and Kyriakides 2007) instead of the Cartesian coordinate system. However, the orientation is given and cannot be (spatially) influenced by the manufacturing process. Hence, the non-constant material orientations are known a priori.

The special case of transversal isotropic material occurs in the context of composite materials and 3D printing based on filaments. There, the material orientations are chosen by the designer. Whereas classic laminates only allow small variations in material orientations, there are already some endless fiber 3D printers based on the fused filament fabrication principle. Additionally, special multi-axial 3D printers allow material orientations in all three dimensions (Kipping and Schüppstuhl 2023; Steltner et al. 2024). Hence, the spatial material orientation is considered as an additional design variable in Lund (2017), Lee et al. (2018), Jantos et al. (2020), and Schmidt et al. (2020), which can also be parameterized using splines (Montemurro et al. 2024). This leads to non-constant and not a priori known anisotropies.

The current contribution aims at giving a decision basis about which multigrid components (especially which smoother) to choose for which type of anisotropic problem. Therefore, different basic approaches are analyzed and benchmarked. Additionally, a new method is proposed to improve the convergence speed. The paper is organized as follows: First, the general multigrid principle is shown. Afterwards, different standard approaches are analyzed using the local Fourier analysis and compared considering computational cost. In the following, these approaches are further investigated by numerical experiments at different (topology optimized) designs. A new approach called spatial Jacobi smoothing is proposed and compared to the previous methods. In the end, some of the methods are applied to different examples of material orientation and topology optimization as well as topology optimization of a wood structure.

2 Solution of the linear equations

The focus of the current publication is to solve the general equation

$$\boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{f}, \tag{1}$$

where $\boldsymbol{K}$ is an arbitrary symmetric and sparse matrix originating from a discretized partial differential equation, $\boldsymbol{u}$ is the state and $\boldsymbol{f}$ is an arbitrary vector.

Using a direct solver, the solution of (1) is found after approximately $\mathcal{O}(n^{3/2})$ operations, where n represents the number of finite elements (Davis 2006). While this is a good performance for small models, the computation time and memory demands become infeasible for large models.

In contrast, iterative methods such as the conjugate gradient method (Hestenes and Stiefel 1952) and the generalized minimal residual method (GMRES) (Saad and Schultz 1986) are mainly based on vector–matrix products. The memory requirements are very low, while the computation time depends on the number of iterations. It is known that the required number of iterations for reaching a given tolerance depends on the condition number, which can be improved by a preconditioner. One very efficient preconditioner is the multigrid method described in Stüben and Trottenberg (1982), Fulton et al. (1986), Wesseling (1992), Briggs et al. (2000), and Trottenberg et al. (2007).

2.1 General multigrid procedure

The basic concept of the multigrid method is to combine the advantages of basic iterative methods with a solution on a coarse mesh. Simple iterative solvers/smoothers, such as the damped Jacobi method are known to reduce high-frequency error components very effectively while low-frequency components remain. In difference, the solution on a coarse mesh only removes low-frequency errors since high-frequency errors are not represented there. Combining both components leads to the multigrid method, where both, low- and high- frequency errors are removed. It has been shown that the multigrid solver only requires $\mathcal{O}(n)$ operations for the Poisson equation if all components are chosen properly (Trottenberg et al. 2007) and hence, it is much more effective than direct solvers for large problems. However, if the components are not chosen properly, the convergence rate suffers. Using the multigrid method as a preconditioner for another iterative method, this issue is reduced.

The overall framework of using multigrid as preconditioner is depicted in Fig. 2. Using the V-cycle, the procedure starts with an initial guess of the solution u_1 (e.g., a zero vector) and the right-hand side f. Inside the outer solver function, the residual is computed. This residual is forwarded to the multigrid preconditioner. On the finest level, a smoothing operator $smooth(\boldsymbol{f}_1)$ (e.g., Jacobi method) is applied and the resulting residual $\boldsymbol{r}_1$ is restricted to a coarser mesh using the restriction operator $\boldsymbol{R}$. The same procedure is repeated on the following meshes until the coarsest mesh is reached (mesh 3 in the figure). On the coarsest mesh, the system of equations is solved exactly (e.g., using direct methods or another iterative solver). Note that the size of the problem is significantly reduced on the coarsest mesh. In the following, the coarse grid solution is prolonged to a finer mesh using a prolongation operator $\boldsymbol{P}$. The result is smoothed on the fine mesh and again prolonged to a finer mesh. This procedure is repeated until the finest mesh is reached. The result of the finest mesh is then transferred to the outer iterative solver (e.g., conjugate gradient). Using the W-cycle (also shown in Fig. 2), the framework is similar. However, the cycle includes additional restrictions after a prolongation.

2.2 Local Fourier analysis (LFA)

The local Fourier analysis is a simple analytic tool to compute the local smoothing properties of different basic iterative methods (such as Jacobi smoothing). The basic concept of the local Fourier analysis is to consider a discretized partial differential equation with constant coefficients (e.g., material properties) and assume an infinite grid (i.e., no effect of boundary conditions) with constant spacing h. Using eigenfunctions, the amplification of different components of the residuum caused by the smoothing operator is computed analytically. Since only local effects are studied, the assumption of

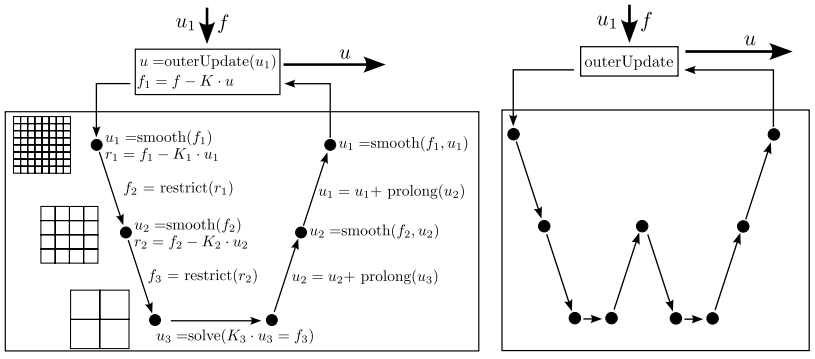


Fig. 2 Framework of a multigrid preconditioned iterative solver. The indices represent the corresponding grid level. Left: V-Cycle, right: W-Cycle

(locally) constant coefficients is also fulfilled for smoothly varying coefficients. A detailed introduction is found in Wienands and Joppich (2004) and Trottenberg et al. (2007). The general error of the state $\boldsymbol{v} = \boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{true}}$ at the position $\boldsymbol{x} + \Delta\boldsymbol{x}$ is represented by an exponential series $\boldsymbol{v}(\boldsymbol{x} + \Delta\boldsymbol{x}) = \sum_j \hat{\boldsymbol{v}}_j(\boldsymbol{x}) e^{i\boldsymbol{\Theta}_j^T \Delta\boldsymbol{x}/h}$ with the grid spacing h and the continuous vector valued error-frequency

$$\boldsymbol{\Theta}_j = \begin{pmatrix} \Theta_{x,j} \\ \Theta_{y,j} \\ \Theta_{z,j} \end{pmatrix} \in \begin{pmatrix} [-\pi, \pi] \\ [-\pi, \pi] \\ [-\pi, \pi] \end{pmatrix} \quad . \tag{2}$$

During the smoothing process, the error of the state is modified by the smoothing operator $smooth(\boldsymbol{v})$. In many cases (e.g., Jacobi), smoothing does not cause any error mixing. This means that smoothing of an error component $\hat{\boldsymbol{v}}_j e^{i\boldsymbol{\Theta}_j^T \Delta\boldsymbol{x}/h}$ leads to an error with the same frequency but a different magnitude. Using local Fourier analysis, for every allowed frequency $\boldsymbol{\Theta}$ the corresponding amplification matrix function $\boldsymbol{S}_h(\boldsymbol{\Theta})$ is computed, such that

$$smooth(\hat{\boldsymbol{v}}_j e^{i\boldsymbol{\Theta}_j^T \Delta\boldsymbol{x}/h}) = \boldsymbol{S}_h(\boldsymbol{\Theta}_j) \hat{\boldsymbol{v}}_j e^{i\boldsymbol{\Theta}_j^T \Delta\boldsymbol{x}/h} \tag{3}$$

and following

$$smooth(\boldsymbol{v}) = \sum_j \boldsymbol{S}_h(\boldsymbol{\Theta}_j) \hat{\boldsymbol{v}}_j e^{i\boldsymbol{\Theta}_j^T \Delta\boldsymbol{x}/h} \quad . \tag{4}$$

For background to the practical computation, the reader is referred to Trottenberg et al. (2007). In the context of the multigrid method, low-frequency errors are corrected by the coarse grid operations, while high-frequency errors must be corrected by the smoothing algorithm. The error is classified as low-frequency if $\boldsymbol{\Theta} \in \begin{pmatrix} [-\frac{\pi}{2}, \frac{\pi}{2}] \\ [-\frac{\pi}{2}, \frac{\pi}{2}] \\ [-\frac{\pi}{2}, \frac{\pi}{2}] \end{pmatrix}$, which means that the wavelength of the error is higher than one element size and hence, seen on the coarse grid. The smoothing factor is then defined as

$$\mu_{loc} = \sup_{\boldsymbol{\Theta}} (\|\rho(\boldsymbol{S}_h(\boldsymbol{\Theta}))\| \, : \, \Theta \in [-\pi, -\frac{\pi}{2}] \cup [\frac{\pi}{2}, \pi]) \tag{5}$$

with $\rho(\boldsymbol{S}_h(\boldsymbol{\Theta}))$ being the spectral radius of the matrix $\boldsymbol{S}_h(\boldsymbol{\Theta})$. In the following, $\rho(\boldsymbol{S}_h(\boldsymbol{\Theta}))$ is called the amplification factor, since it represents how much an error with the frequencies $\boldsymbol{\Theta}$ is amplified. Since the high-frequency error should be reduced, the amplification factor must be lower than 1 for all high frequencies, and hence, the smoothing factor μ_{loc} must be lower than 1.

Application of the local Fourier analysis to the Jacobi method with the standard isotropic material gives the amplification factors shown in Fig. 3. There, the frequency in the z-direction is set to zero, while the amplification factor is given for different frequencies in the x- and y-directions. The white part of the surface plot indicates the low-frequency domain and the green part indicates the high-frequency domain. For all frequencies, the amplification factor stays below one, meaning that the Jacobi method is also stable as a standalone solver. In general, the amplification factors are lower in the relevant high-frequency domain. There, the worst amplification factor is found at the border to the low frequency domain with $\mu_{loc} \approx 0.75$.

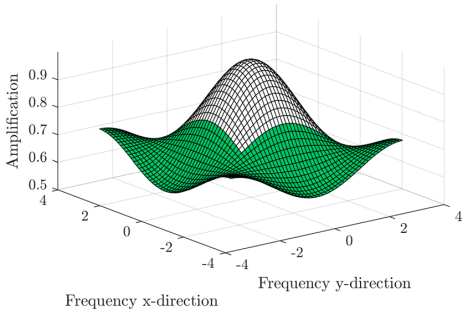


Fig. 3 Amplification factors from the local Fourier analysis of Jacobi smoothing with $\omega = 0.5$ for isotropic material. The x- and y-axis represent the frequency of the error of the state in x-/y-direction. The z-axis represents the corresponding amplification factor. For visualization reasons, the error frequency in z-direction is set to 0

3 Theoretic analysis of common smoothing methods for orthotropic materials

Besides the Jacobi method, there are many other common smoothing algorithms, such as the Gauß-Seidel method, line-smoothers and plane-smoothers. The Gauß-Seidel method is known to be more robust and shows better smoothing properties than the Jacobi method. However, it is known to show similar behavior to the Jacobi method for anisotropic partial differential equations. Alternatively, line-smoothers and plane smoothers are recommended for anisotropic problems in literature, but have not been tested for the special case of topology optimization and non-constant orientations. In general, all alternatives to the Jacobi method are more computational costly and complex to implement efficiently. Hence, a fair comparison does not only include the required number of solver iterations, but also the (theoretical) computational effort to apply one smoothing step as well as the parallelizability. These methods are shortly explained and analyzed using local Fourier analysis. Afterwards, the methods are compared regarding smoothing properties for different material orientations and computational cost. As exemplary material, Balsa wood with the mechanical

properties shown in Appendix D.2 is considered, since it is a commonly used wood and shows highly orthotropic behavior. However, the results can be generalized to other materials with comparable stiffness ratios and transversal isotropic materials, such as carbon fiber reinforced polymers.

In the current contribution, the (spatially varying) material orientation of the orthotropic material is described by the cardan angles as depicted in Fig. 4. In the original coordinate system, the orthotropic material has its highest stiffness in the X-direction and the lowest stiffness in the Y-direction. The first rotation η acts around the X-axis, the second rotation φ acts around the new z-axis and the third rotation θ acts around the new y-axis. Note that the specific description of the material orientation is only required for practical generation of the material tensor. The performance of the multigrid components is only influenced by the resulting discretized partial differential equation i.e., the (tangential) stiffness matrix, and not by the specific representation of the material orientation.

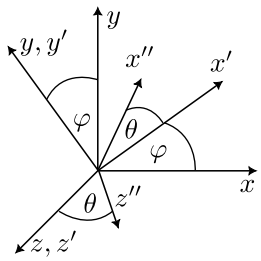


Fig. 4 Rotations of the transversal isotropic material. For the orthotropic material, an additional rotation η around the x-axis is done before. In the initial system, the material has the highest stiffness in x-direction and the lowest stiffness in the y-direction

3.1 Jacobi method

Due to its simplicity and low computational cost, the damped Jacobi method is often used as a smoother for the multigrid method (Amir et al. 2014; Aage et al. 2015; Peetz and Elbanna 2021). The response update $\boldsymbol{u}_{i+1}$ is computed by

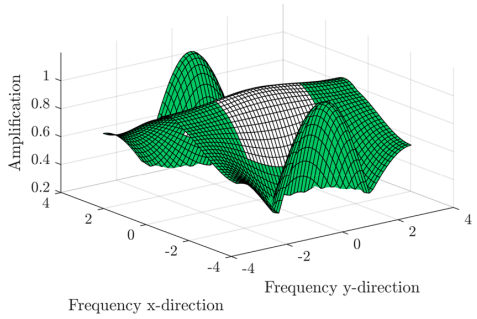
$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \omega \boldsymbol{D}^{-1} \cdot \left(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}_i\right), \tag{6}$$

where $\boldsymbol{D}$ represents the diagonal of the system matrix $\boldsymbol{K}$. As discussed in the introduction, the Jacobi method substantially suffers from orthotropic material behavior. The reason for the poor performance in the context of orthotropic material is further analyzed using local Fourier analysis (LFA).

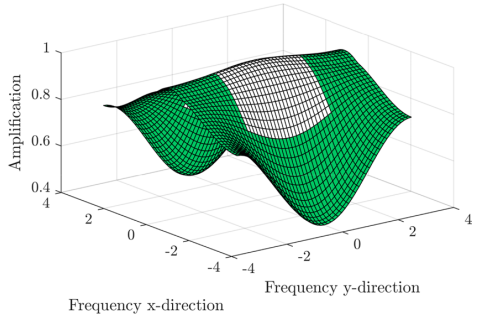
Considering only 0° orientations and using the damping factor $\omega = 0.5$, the amplification factors depicted in Fig. 5a are found. Compared to the example with isotropic material (Fig. 3), the amplification factors are much higher. Especially the peaks for high frequencies in the x-direction indicate that the method is unstable. However, also in y-direction, the amplification factor is stuck slightly below 1. When using $\omega = 0.3$ instead, the amplification shown in Fig. 5b is observed. The previous unstable regions now show good performance with amplification factors below 0.9. However, the smoothing in the y-direction gets even worse with $\mu_{loc} \approx 0.98$. Hence, it is crucial to choose an appropriate damping factor to obtain a stable method with acceptable smoothing properties.

3.2 Decoupled point-Gauß-Seidel smoothers/ successive overrelaxation

The decoupled pointwise successive overrelaxation (SOR) is very similar to the damped Jacobi method. However, it



(a) $\omega = 0.5$



(b) $\omega = 0.3$

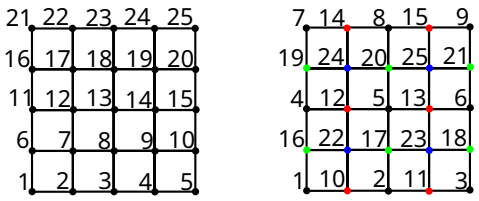
Fig. 5 Amplification factors from the local Fourier analysis of Jacobi smoothing for Balsa wood. The x- and y-axis represent the frequency of the error of the state in x-/y-direction. The z-axis represents the corresponding amplification factor. For visualization reasons, the frequency in z-direction is set to 0

solves a triangular instead of a diagonal system of equations. The state update reads

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \left(\boldsymbol{L} + \frac{1}{\omega}\boldsymbol{D}\right)^{-1} \cdot \left(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}_i\right) \tag{7}$$

with the strict lower triangular part $\boldsymbol{L}$ of the system matrix $\boldsymbol{K}$. The Gauß-Seidel method is the special case when using $\omega = 1$, whereas the method is called successive overrelaxation for $\omega > 1$. The smoothing properties depend on the node numbering of the finite element mesh. For a structured mesh, there are two major numbering groups: lexicographic numbering and the multicolor numbering. Similar numberings are also found for unstructured meshes. In the lexicographic numbering, the nodes are numbered linewise, such that neighboring nodes have following numbers (see Fig. 6a). If a line is finished, the first node of the next line has the following number. In difference, the multicolor numbering sorts the nodes into different color groups (see Fig. 6b). The groups are defined such that neighboring nodes belong to different colors. The node numbers are counted for each color separately. In Fig. 6b , black nodes have the numbers 1–9, red nodes the numbers 10–15, green nodes the numbers 16–21 and blue nodes the numbers 22–25. Using the multicolor numbering, the smoothing algorithm can be implemented in parallel without high implementation effort. A simple parallel formulation is explained in Appendix A. In difference, the lexicographic version is difficult to parallelize.

Applying the local Fourier analysis to the lexicographic Gauß-Seidel method for linear elasticity with Balsa wood (same orientation as before), the amplification factors shown in Fig. 7 are found. As for the Jacobi method, the amplification factor is stuck slightly below 1 in the y-direction, while good smoothing properties are obtained in the x-direction. However, no stability issues occur, and the absolute values are significantly better, leading to an overall smoothing factor $\mu_{loc} \approx 0.9$ instead of 0.98.



(a) Lexicographic numbering. (b) Multicolor numbering.

Fig. 6 Different node numbering schemes for the successive overrelaxation

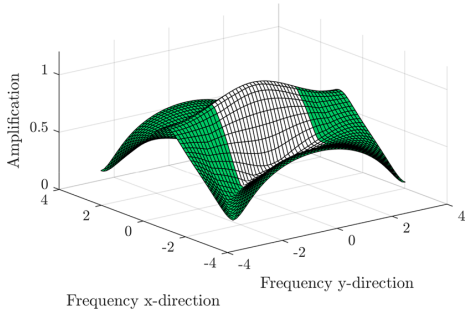


Fig. 7 Amplification factors from the local Fourier analysis of lexicographic Gauß-Seidel smoothing for Balsa wood. The x- and y-axis represent the frequency of the error of the state in x-/y-direction. The z-axis represents the corresponding amplification factor. For visualization reasons, the frequency in z-direction is set to 0

3.3 Coupled point successive overrelaxation

The coupled pointwise successive overrelaxation is similar to the decoupled successive overrelaxation. Again, the status update reads

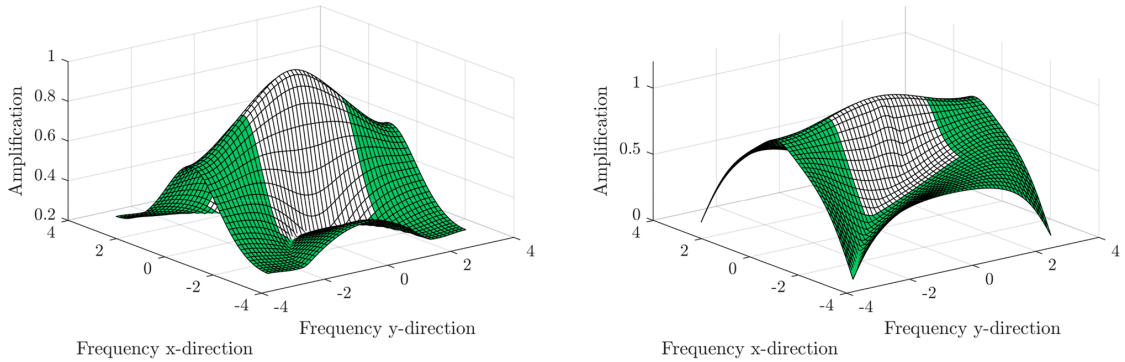
$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \left(\boldsymbol{L}_p + \frac{1}{\omega}\boldsymbol{D}_p\right)^{-1} \cdot \left(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}_i\right) \quad , \tag{8}$$

however here, the matrix $\boldsymbol{D}_p$ is a block diagonal matrix that contains couplings between different degrees of freedom of the same node. The lower triangular matrix $\boldsymbol{L}$ is modified such that $\boldsymbol{K} = \boldsymbol{L}_p + \boldsymbol{D}_p + \boldsymbol{L}_p^T$. If all rotations are set to 0, no couplings occur between the degrees of freedom of the same node, and hence, the coupled successive overrelaxation is equal to the decoupled version. Similar to the decoupled SOR method, the smoothing properties depend on the node numbering scheme. However, due to the block diagonal structure of $\boldsymbol{D}_p$, lexicographic smoothing results in an infeasible computational cost.

3.4 Line/plane relaxation methods

An alternative way to improve the smoothing properties of a multigrid method is to use block iterative methods (Arms et al. 1956) such as block Successive Overrelaxation (similar to Gauß-Seidel) (Evans and Biggins 1982) or the block Jacobi method (Ferguson 1986). In the context of anisotropic partial differential equations, line successive overrelaxation and plane successive overrelaxation are popular choices (Trottenberg et al. 2007). Equal to the case of coupled point SOR, the status update reads

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \left(\boldsymbol{L}_{L/P} + \frac{1}{\omega}\boldsymbol{D}_{L/P}\right)^{-1} \cdot \left(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}_i\right) \quad ,, \tag{9}$$



(a) Lines oriented into fiber direction. (b) Lines perpendicular to fiber direction.

Fig. 8 Amplification factors from the local Fourier analysis of line SOR smoothing with $\omega = 1$ for Balsa wood. The x- and y-axis represent the frequency of the error of the state in x-/y-direction. The z-axis represents the corresponding amplification factor. For visualization reasons, the frequency in z-direction is set to 0

where the matrix $\boldsymbol{D}_{L/P}$ is a block diagonal matrix that contains couplings between different degrees of freedom of the same box (line/plane). The lower triangular matrix $\boldsymbol{L}_{L/P}$ is chosen such that $\boldsymbol{K} = \boldsymbol{L}_{L/P} + \boldsymbol{D}_{L/P} + \boldsymbol{L}_{L/P}^T$. When choosing the lines that are oriented in the same direction as the stiffest material direction, substantial improvements in the smoothing are obtained. Application of the local Fourier analysis to x-line SOR for Balsa wood with $\eta = 0, \varphi = 0, \theta = 0$ shows completely different behavior to the decoupled SOR method. Figure 8a shows the amplification factors for different frequencies in the x- and y-directions. In difference to classic lexicographic SOR (as shown in Fig. 7), line SOR also shows good smoothing properties for high frequencies in the y-direction. In consequence, the overall smoothing factor is much better with $\mu_{loc} < 0.8$ instead of 0.9. If, however, the lines are not oriented into the stiffest material direction, no advantage is found compared to coupled successive over-relaxation (See Fig. 8b). Similar observations are done for plane SOR.

For both relaxation algorithms, again, the smoothing properties depend on the node numbering. When using the multicolor numbering, the colors contain lines/planes that do not have any connections to each other. In this case it is sufficient to approximate the solution of (9) using one V-cycle of the multigrid method for every plane/line (Thole and Trottenberg 1986) separately. Alternatively, the system of equations could be solved using conjugate gradients. Own investigations showed that a desired tolerance of less than 10^{-2} often does not give any improvements for the global multigrid solve.

3.5 Direct comparison of smoothing methods

Besides computing the amplification factor function for a single orientation, the local Fourier analysis is also able to compute the smoothing factors for several orientations. Figure 9 shows smoothing factor maps for different smoothing algorithms. There, the smoothing factors are depicted for all rotations φ and θ. The rotation η is set to zero for visualization reasons. Additionally, Table 1 presents the minimum, maximum and mean value as well as the orientations of best/worst performance for all considered methods.

For the Jacobi method, the damping factor $\omega = 0.3$ is chosen since it is an appropriate value for $\varphi = 0, \theta = 0$. However, instabilities occur for several orientations where both angles are rotated by approximately $\pm 15°$. Using $\omega = 0.25$ instead, unstable regions are removed at the cost of slightly worsened smoothing factors at the other orientations. The best values are found at orientations that align to the coordinate axes since these cases contain no couplings between degrees of freedom at the same node. There, the smoothing factor reaches values around 0.98.

Like the Jacobi method, the decoupled lexicographic SOR method shows orientation-dependent smoothing properties. Considering $\omega = 1$ (i.e., Gauß-Seidel method) the qualitative appearance of good and poor performance is similar to the Jacobi method. However, the absolute quantities are significantly better, with values between 0.84 and 0.96. The best performance is obtained for orientations that are near $\varphi = 0, \theta = 0$. This is explained by the node-numbering-dependent properties. In the current case, the nodes are numbered line-wise with the lines oriented in the x-direction. This means that a strong coupling in x-direction is implemented in the smoothing process. Hence, good

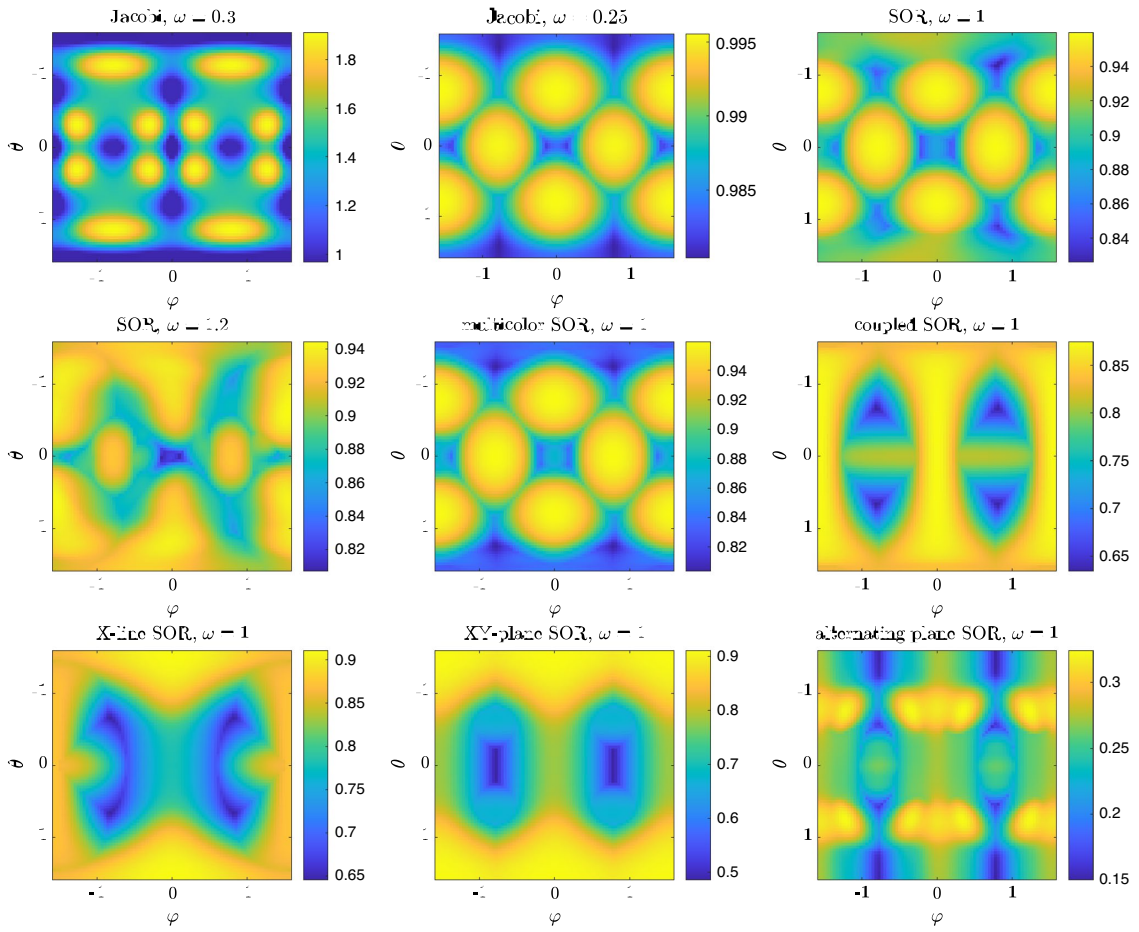


Fig. 9 Smoothing factors μ_{loc} for different smoothing algorithms considering Balsa in different orientations φ, θ computed using the local Fourier analysis. The first rotation around the x-axis is set to zero, i.e., $\eta = 0$. The colors indicate the smoothing factors values

Table 1 Smoothing factors and orientations with best or unstable behavior for different smoothing algorithms obtained using local Fourier analysis

Smoothing method	Smoothing factor			Best/unstable
	Minimum	Maximum	Mean	Orientation
Jacobi $\omega = 0.3$	0.976	1.18	1.01	Unstable at $\approx \pm 15°$
Jacobi $\omega = 0.25$	0.98	0.996	0.99	Coordinate axes
SOR $\omega = 1$	0.83	0.96	0.92	x-direction
SOR $\omega = 1.2$	0.81	0.94	0.91	x-direction
Multicolor SOR $\omega = 1$	0.80	0.96	0.90	Coordinate axes
Coupled SOR $\omega = 1$	0.63	0.87	0.81	$\approx \pm 45°$
X-line SOR $\omega = 1$	0.65	0.91	0.81	x-direction
XY-plane SOR $\omega = 1$	0.49	0.91	0.78	x-y-plane
Alternating plane SOR $\omega = 1$	0.15	0.32	0.26	All directions

smoothing properties are obtained for the stiffest direction pointing in x-direction. If the numbering was line-wise with

lines oriented into the y-direction, the best smoothing properties are obtained for $\varphi = 90°, \theta = 0$. Increasing the damping factor to $\omega = 1.2$, this effect is even amplified, leading to superior smoothing factors at $\varphi = 0, \theta = 0$ and inferior factors for large rotations. However, considering all orientations, the mean smoothing factor is improved by less than 1%. Higher damping factors do not improve the performance anymore, but lead to unstable methods.

In difference to the lexicographic ordering, the multicolor ordering does not have a directed numbering and hence, also does not show superior behavior for a certain coordinate axis direction. Besides this effect, the behavior is nearly equal to the lexicographic version.

Considering couplings at single nodes by utilization of the coupled multicolor SOR smoother with $\omega = 1$ improves the performance for orientations that are not aligned with the coordinate axes. Especially at orientations where both rotations are near ±45°, coupled multicolor SOR shows its best performance with smoothing factors around $\mu_{loc} = 0.65$. This comes from the fact that the coupling between the degrees of freedom of a single node reaches its maximum for rotations of approximately ±45°.

By using line or plane smoothing, superior smoothing factors are obtained for orientations that align with the line or plane. There, the smoothing factor is improved by more than 15% compared to coupled smoothing. Since line and plane smoothing also include all couplings at the same node, they also work well for all angles, where the coupled SOR method worked well. In consequence, the smoothing factor is less than 0.8 for many angles. However, as expected, the performance is rather bad for material orientations that are perpendicular to the smoothing line or plane (e.g., $\theta \approx \pm 90°$).

A more robust formulation is the alternating plane smoothing. There, plane smoothing is done for the x-y, y-z and x-z planes sequentially. This obviously comes at the cost of three smoothing swipes of single-plane smoothing. However, application of the local Fourier analysis reveals that alternating plane smoothing shows good performance for all orientations (including these, where the previous methods performed badly) with smoothing factors of less than $\mu_{loc} < 0.33$. In fact, it performs better than three applications of xy-plane smoothing for many material orientations.

Based on these results, the alternating plane smoothing is clearly the best smoothing approach. However, despite the smoothing factors, the previously mentioned algorithms have substantially different computation times and memory demands. The required number of scalar multiplications and a relative computation time for a naive Matlab implementation are shown in Table 2. The corresponding assumptions and detailed derivations are found in b. Paying attention to the computation times, the ranking depends on the specific orientations. Especially coupled successive overrelaxation and line successive overrelaxation seem to be very competitive since they are much less computationally costly than alternating plane smoothing. However, there are many effects in practical application that are not considered by the local Fourier analysis. Therefore, the smoothing algorithms are applied in the context of practical optimization problems.

4 Numeric analysis of existing smoothing methods

To validate the results of the local Fourier analysis and extend them to real topology optimization problems, different smoothing methods are analyzed regarding their properties when solving the finite element equation. In the following analysis, the required number of iterations is computed for four different designs. In the first experiment, the special case of constant coefficients (i.e., constant density field, constant orientations) is considered, which is as close as possible to the assumptions made for the local Fourier analysis. Afterwards, the more general cases of non-constant orientations and/or non-constant densities are considered. Again, Balsa wood is used as material. For the application of the multigrid solvers to different problems (4.2 to 4.5), the V-Cycle multigrid method is used as a preconditioner to the conjugate gradient method, since this is state of the art in the field of engineering optimization. Very tight solver tolerances of 10^{-10} are considered to get reproducible results, and only one smoothing swipe is applied on each grid. The

Table 2 Required number of multiplications per node for different smoothing algorithms for the case that the global stiffness $\boldsymbol{K}$ is available or not

	Smoothing method					
	Jacobi	SOR	Coupled SOR	Line SOR	Plane SOR	Alternating SOR
$\boldsymbol{K}$ available	~ 250	~ 360	300–500	~ 800	~ 2000	~ 5500
$\boldsymbol{K}$ not available	~ 580	~ 1200	700–850	~ 1100	~ 2500	~ 6000
Observed time Matlab	100%	200%	250%	450%	800%	2400%

The observed time using a naive (sequential) Matlab implementation is given relative to the Jacobi method, where $\boldsymbol{K}$ was available

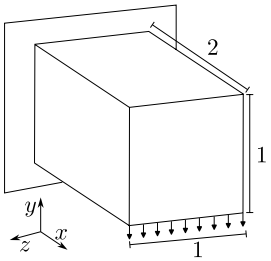
damping factor is set to $\omega = 0.25$ for the Jacobi method, since the local Fourier analysis reveals that higher values lead to unstable behavior. For all other methods, the damping factor is set to $\omega = 1$, since this case corresponds to Gauß-Seidel methods. According to the local Fourier analysis, increasing the value does not cause significantly improved smoothing. The decoupled multicolor SOR method is not considered since it is always inferior to the coupled version at similar computational cost.

4.1 Example considered

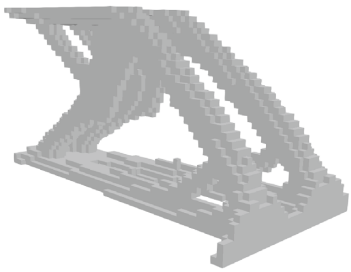
For the analysis examples with non-constant densities, the density field is computed using a topology optimization of a cantilever beam, which is discussed in more detail in Sect. 7.1. The corresponding problem is taken from Peetz and Elbanna (2021).

The design space and the boundaries are depicted in Fig. 10a . The problem consists of a cuboid-shaped design space that is discretized using a structured mesh of $64 \times 32 \times 32$ isometric hexagonal finite elements. The low number of finite elements is chosen such that the direct solve can be applied already at the second grid in order to enable an analysis of the two grid method. The model is clamped on the back surface and loaded by a line load on the lower front edge. For the design generation, isotropic linear elastic material behavior with a Poissons's ratio of 0.3 is considered. The densities are filtered using the PDE filter (Lazarov and Sigmund 2011) with a filter radius of 2 times the element edge length and afterwards penalized using the modified SIMP approach (Sigmund 2007). Like in Peetz and Elbanna (2021), a continuation scheme is applied for the penalization parameter p. Starting from 1, p is increased by 0.25 every 20 iterations up to a maximum value of $p = 4$. At every change of p a new optimization is started using the previous design as the initial guess. Considering an allowed volume fraction of 12%, the optimization using the method of moving asymptotes (Svanberg 1987) leads to the final design shown in Fig. 10b . The result is qualitatively similar to the design gained in Peetz and Elbanna (2021). However, visible differences are caused by a different discretization, filter type

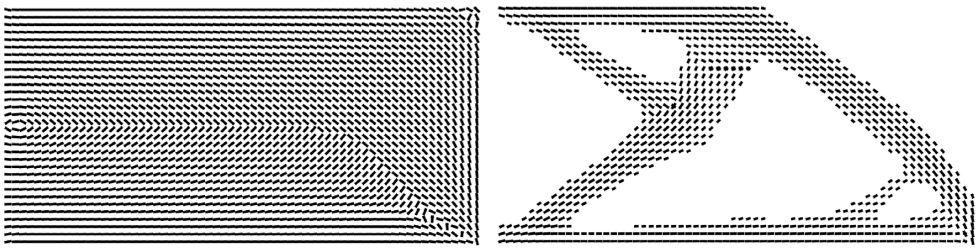
Fig. 10 Compliance topology optimization example of an cantilever used for the following numeric experiments



(a) Design space and boundary conditions for the topology optimization problem.



(b) Final design of the isotropic topology optimization.



(a) Orientation field for the constant density field.

(b) Orientation field for the non-constant density field.

Fig. 11 Side view of the orientation fields for the example problems considering constant and non-constant densities. The orientations are aligned to the first principal stress direction

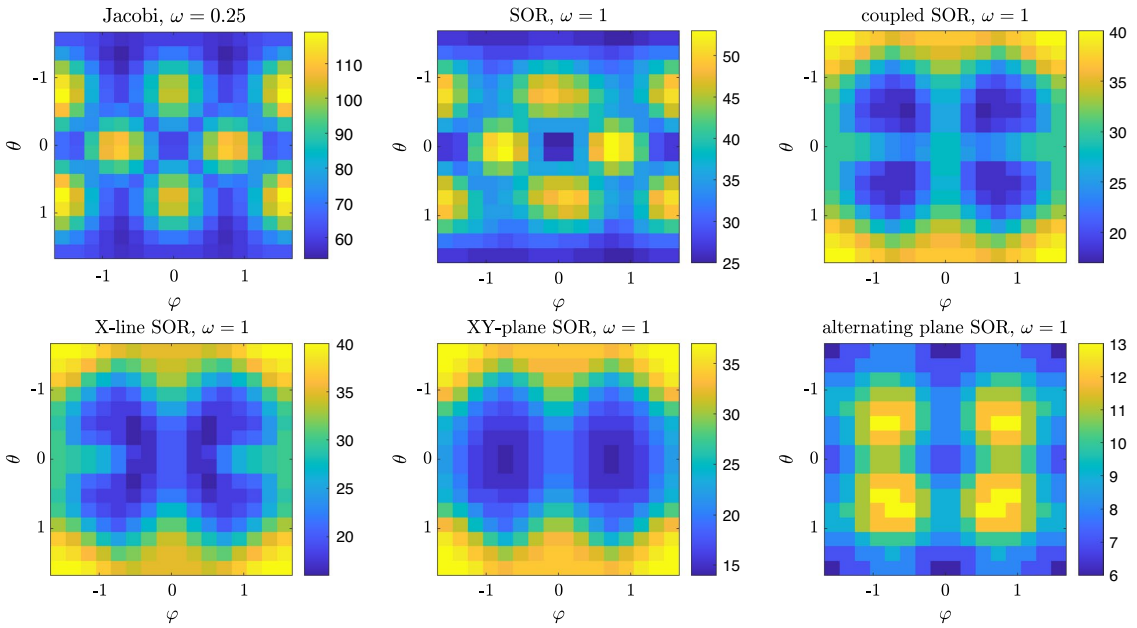


Fig. 12 Required number of iterations for the multigrid method to reach a relative tolerance of 10^{-10} for different constant material orientations φ, θ of Balsa wood. Here, the elasticity equations of the example problem 10a with constant densities is solved using 2 grids. The colors indicate the required number of iterations

and relative filter radius. Especially the very coarse discretization in combination with a low volume fraction leads to many elements with intermediate densities. As previously mentioned, numerical experiments are also done for spatially varying orientations combined with constant and optimized topology. For simplicity, the material is oriented into the principal stress directions. Thereby, the rotation around the x-axis is set to zero, i.e., $\eta = 0$. A 2D-cut of the orientation field is shown in Fig. 11.

4.2 Application of methods to example with constant densities and constant orientations

First, the multigrid solver is applied to the example problem 10a with constant densities and constant orientations. Figure 12 shows the number of iterations[1] needed when using 2 grids, such as assumed in the local Fourier analysis. Qualitatively, the results are equivalent to the local Fourier analysis. For orientations where low smoothing factors were obtained, the number of iterations is also low. Additionally, the ranking of smoothing methods is the same. However, the Jacobi method performs better than expected. For the orientation $\varphi = 0, \theta = 0$ the smoothing factor $\mu_{loc} = 0.98$ was obtained. To reduce the high-frequency error to 90%, which is the smoothing factor of lexicographic SOR, approximately 5 smoothing swipes are required ($0.98^5 \approx 0.9$). However, the experiments only show a difference of factor 2. The improved performance of the Jacobi method is explained by beneficial effects of the outer conjugate gradient solver. In additional numerical experiments, the authors used the multigrid method as a standalone solver. There, the iteration numbers were a factor of 2 to 5 higher for all smoothing methods, and especially the Jacobi method performed approximately 5 times worse than the lexicographic SOR method. In practical application, rather 3–5 grids are used than only 2 grids to reduce the cost of the coarse grid solve (Amir et al. 2014; Peetz and Elbanna 2021). Repeating the same experiment with four grids, the numbers of iterations are similar to the results shown in Fig. 12. The values are increased by up to two in only a very small number of orientations.

[1] Iterations here refer to the solution of the equation system, not to the number of optimization iterations.

Table 3 Number of iterations to compute the deformations of example problem 10a, where the orientations of the Balsa wood are set to the principal stress direction or are spatially random

Example	Multigrid approach	Smoothing method					
		Jacobi	SOR	Coupled SOR	Line SOR	Plane SOR	Alternating SOR
Stress-aligned orientations	2 grids	92	41	29	28	27	8
	4 grids	94	43	31	29	28	9
Random orientations	4 grids	59	32	30	29	28	14

The V-Cycle procedure is used with the desired tolerance 10^{-10}

4.3 Application of methods to example with constant densities and optimized orientations

Considering the orientation field presented in Fig. 11a in combination with constant densities, the iteration numbers represented in Table 3 are obtained. There, the two grid cycle as well as the four-grid cycle are applied. The achieved numbers of iterations are approximately the mean of the iterations required for all included orientations. Hence, the non-constant orientations do not cause extra problems for the multigrid solver. In fact, there is an opposite effect. Possible bad orientations in the design are balanced by good orientations leading to an intermediate performance. The same effect is observed in the case of uncorrelated random orientations. There, the balancing effect is even more present. Also, using multiple grids does not worsen the convergence speed.

4.4 Application of methods to example with optimized topology and constant orientations

Application of the multigrid method to the topology optimized design shown in Fig. 10b with constant material orientations and two grids leads to the iteration numbers shown in Fig. 13. The general relationship of iteration number and orientation is similar to the case of constant densities, but the absolute numbers increase. While the iteration numbers are only increased by 10%-20% for Jacobi and lexicographic SOR, the numbers are approximately doubled for the multicolor methods, still showing better performance. This is caused by two reasons. First, the problem of anisotropy seems to be rather independent of the problem of high contrasts. Hence, the number of iterations is increased by approximately the same amount for all methods, which leads to a higher relative increase for the methods that performed better for constant densities. Second, the multicolor pattern strictly requires the low-frequency errors to be suppressed by the coarse grid solve. For more details, see the derivation of the local Fourier analysis in Trottenberg et al. (2007). In the context of high contrast designs, this requirement is not fulfilled anymore, since structural details of the fine grid are not properly represented on the coarse grid.

In difference to the case of constant densities, the multicolor smoothers also suffer from an inaccurate solve of the state update Eq. 9. Using only one swipe of the corresponding 1D/2D multigrid method even leads to divergence of the overall solver. This is caused by poor performance of the multigrid method in the case of high-contrast designs on more than two grids (Peetz and Elbanna 2021). However, solving the line/plane problem with an accuracy of at least 10^{-2} leads to less than 10% additional iterations for the overall solve.

When using the multigrid method with four grids instead of two grids, the numbers of iterations totally explode for some orientations. Instead of 30–130 iterations, 300–450 iterations are required. The corresponding figure is found in c. However, numerical experiments show that a similar loss of performance is also found for isotropic materials, such as steel, where the number of iterations grows by a factor of 5. Hence, it is expected that the reasons are equal to the case of isotropic topology optimization (e.g., discussed in Amir et al. (2014); Peetz and Elbanna (2021)). There, the major problem is that the details of the fine grid are not properly represented on the coarse grid.

A simple fix to improve the convergence of the multigrid method with many grids and topology optimized designs is to use several smoothing steps on the coarse grids or to apply a different multigrid cycle. Since the convergence is good when using only two grids, the problems must be caused by the coarse grids. In consequence, improving the components at coarse grids also improves the overall performance.

When using the W-Cycle instead of the V-Cycle, the computational cost is increased by approximately 15% per iteration. Application of the W-Cycle to the example with optimized topology with four grids leads to the iteration numbers presented in Fig. 14. Compared to the V-Cycle, the number of iterations drops by 30% to 50%, which is still significantly more than in the case with 2 grids. Additionally,

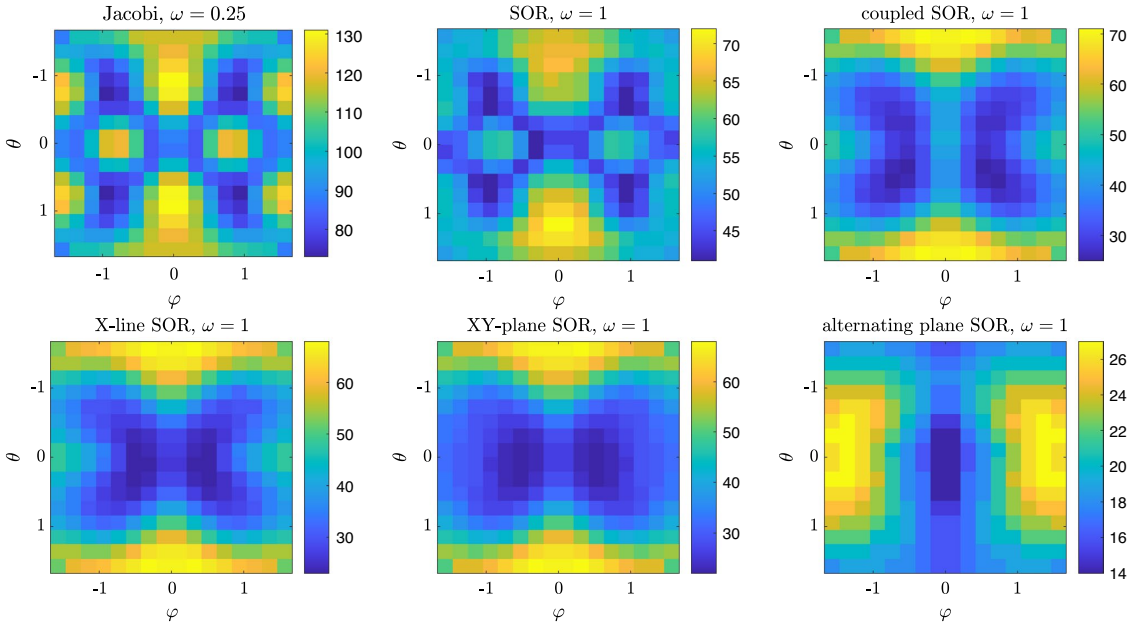


Fig. 13 Required number of iterations for the multigrid method to reach a relative tolerance of 10^{-10} for different constant material orientations φ, θ of Balsa wood. Here, the elasticity equations of the example problem 10b with optimized densities is solved using 2 grids. The colors indicate the required number of iterations

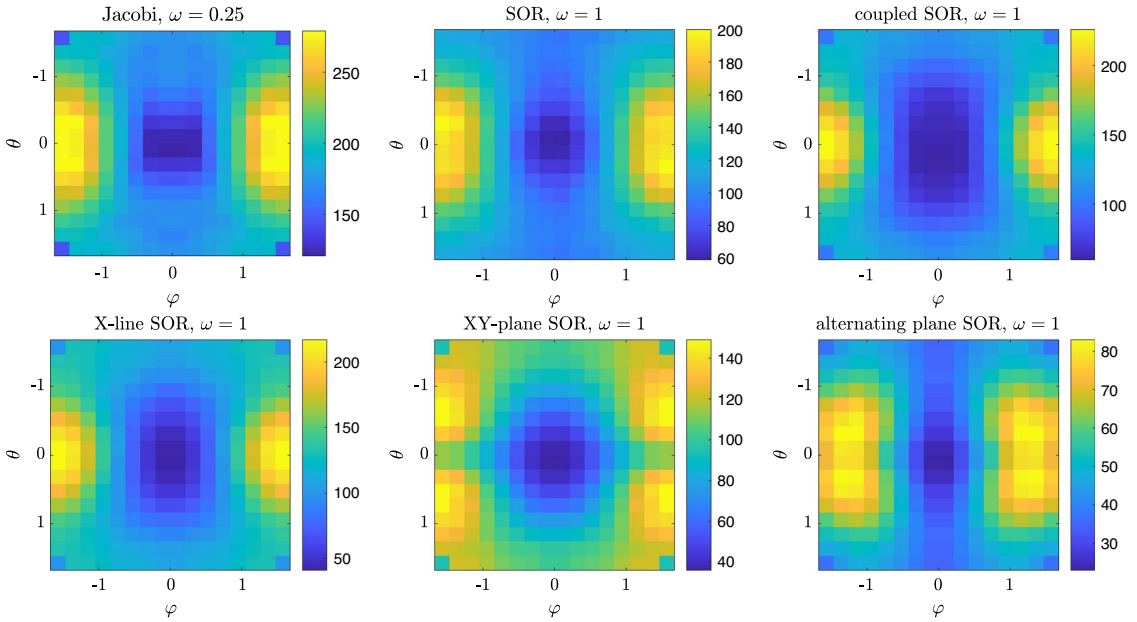


Fig. 14 Required number of iterations for the multigrid method to reach a relative tolerance of 10^{-10} for different constant material orientations φ, θ of Balsa wood. Here, the elasticity equations of the example problem 10b with optimized densities is solved using the W-Cycle with 4 grids. The colors indicate the required number of iterations

the general shape of the iteration number plot is fundamentally changed compared to the case with 2 grids. Especially orientations with $\varphi = 0, \theta = 0$ now show best performance, which might be explained by the fact that the shape of the topology is also oriented to the x-axis.

4.5 Application of methods to example with optimized topology and varying orientations

Combining non-constant densities with stress-aligned orientations shown in Fig. 11b leads to the iteration numbers presented in Table 4. There, all previously discussed cases, i.e., two grids, four grids and four grids with W-cycle, are considered. For comparison, also the number of iterations for spatially random orientations and isotropic steel as material are shown.

In the case of two grids, the number of iterations is similar to the case with constant densities. Considering optimized orientations, the number of iterations does not explode when going to four grids. In fact, the values only increase by less than 25%. Again, good orientations balance bad orientations, leading to overall good performance. Unexpectedly, this means that the required number of iterations is even lower than when considering isotropic steel as material. The reason is found in the specific orientation field. Since the material is oriented into the principal stress direction, the principal material direction aligns with the structural elements. The stiffness in fiber direction is much higher than the orthogonal stiffness and the shear stiffness. In consequence, small errors in deformations at the transition zones have less influence on the force residual. This effect does not occur in the context of completely random orientations. There, only the balancing effect between good and bad orientations is observed, leading to worse values than for the isotropic material. When using the W-cycle instead of the V-cycle for multiple grids, again, the number of iterations drops significantly.

Table 4 Number of iterations to compute the deformations of the topology optimized example problem 10b, where the orientations of the Balsa wood are set to the principal stress direction or are spatially random

Example	Multigrid approach	Smoothing method					
		Jacobi	SOR	Coupled SOR	Line SOR	Plane SOR	Alternating SOR
orientation stress-aligned	2 grids	100	48	29	28	28	11
	4 grids	104	56	36	34	34	21
	W-Cycle 4 grids	102	50	31	29	28	15
random orientations	4 grids	266	185	184	185	161	99
isotropic	4 grids	130	97	77	77	70	49

The desired solver tolerance is set to 10^{-10}

Fig. 15 Algorithm for computing optimal damping factors in the context of spatial Jacobi smoothing. This preprocessing must be done only once for every material

```
Generate vectors of angles η = [−π/2 ... π/2]^T, φ = [−π/2 ... π/2]^T, θ = [−π/2 ... π/2]^T
Generate vector of damping factors ω = [0 ... 1]^T
Initialize lookup table ω_opt
for i_η
  for i_φ
    for i_θ
    Initialize smoothing factor vector μ_loc
      for i_ω
        Compute smoothing factor μ_loc,i_ω = μ_loc(η_i_η, φ_i_φ, θ_i_θ, ω_i_ω) using LFA
      end
      Find index i_opt of lowest value of μ_loc
      ω_opt,η_i_η,φ_i_φ,θ_i_θ,ω_i_ω = ω_i_opt
    end
  end
end
Compute intermediate values of ω_opt using a response surface
```

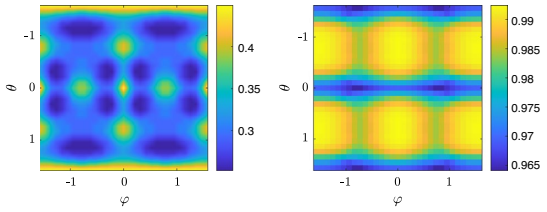
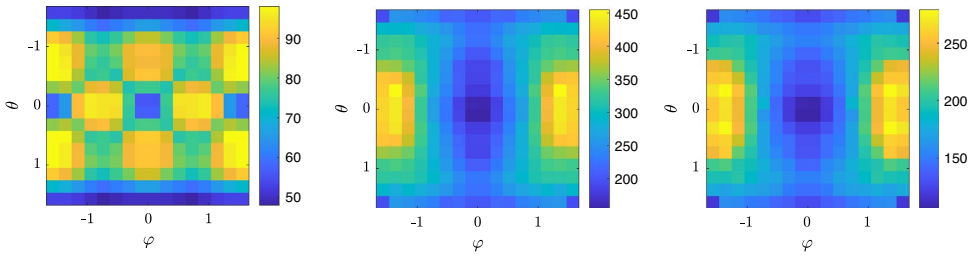


Fig. 16 Optimal damping factor ω (left) and resulting smoothing factors μ_{loc} (right) for Jacobi smoothing with Balsa for different orientations. The values are obtained using local Fourier analysis. The colors indicate the optimal damping factors (left) and the resulting smoothing factors (right). The angle η is set to zero for visualization reasons

5 New approach: spatial smoothing

The previous analyses showed that the (decoupled point) Jacobi method does not perform as well as coupled or box smoothers. However, especially in the context of topology optimized designs, the difference is not substantial (factor 2–4). On the other hand, the Jacobi method is simple, easy to implement, and very fast in the evaluation. The most crucial component is to choose the damping factor ω properly. In difference to the SOR schemes, where $\omega = 1$ is stable for all situations, the Jacobi method is stable for different damping factors for different materials and orientations. In Luo et al. (2024) it is proposed to compute optimal damping factors using an eigenvalue analysis of the true iteration matrix. This leads to good performance of the solution process, but it requires the expensive solution of an eigenvalue problem.

In the current publication, we propose a similar approach. However, the focus lies on different materials with spatially varying orientations. Hence it is proposed to use different damping factors for every node instead of one global value. Therefore, “optimal" damping factors are computed for all possible material orientations using the local Fourier analysis and stored in a lookup table. These values are only computed once for every material and do not change. During the (material orientation and) topology optimization, the optimal damping factors are extracted from the lookup table for every element separately. Afterwards, the nodal damping factors are computed from the elemental values by taking the volume-weighted average of all surrounding elements. Finally, the spatial damping factors are applied to the Jacobi method by



(a) Initial design, V-Cycle. (b) Optimized design, V-Cycle. (c) Optimized design, W-Cycle.

Fig. 17 Required number of iterations for the multigrid method using spatial Jacobi smoothing to reach a relative tolerance of 10^{-10} for different constant material orientations φ, θ of Balsa wood. Here, the elasticity equations of the example problems 10b and a are solved using 4 grids. The colors indicate the required number of iterations

Table 5 Number of iterations to compute the deformations of example problem 10a, where the orientations of the Balsa wood are set to the principal stress direction

Orientation field	Density field	Cycle type	Smoothing method		
			Spatial Jacobi	Jacobi with $\omega = 0.25$	Improvement (%)
Stress-aligned orientations	Initial design	V	76	94	20
	Optimized design	V	86	104	17
	Optimized design	W	82	102	20
Random orientations	Optimized design	V	248	266	7
	Optimized design	W	147	163	10

Four grids and a relative tolerance of 10^{-10} are applied

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \boldsymbol{\omega D}^{-1} \cdot \left(\boldsymbol{f} - \boldsymbol{K u}_i\right) \tag{10}$$

with

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_1 & 0 & 0 & \dots \\ 0 & \omega_2 & 0 & \dots \\ 0 & 0 & \omega_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad . \tag{11}$$

A pseudo-code for computing the optimal factors is shown in Fig. 15. First, linearly spaced realizations of angles $\eta, \varphi, \theta \in [-\frac{\pi}{2}, \frac{\pi}{1}]$ and damping factors $\omega \in [0, 1]$ are generated. Afterwards, the local Fourier analysis is applied to every combination of angles and damping factors. The damping factors leading to the best local smoothing factor for each combination of angles is stored as the corresponding optimal damping factor. After the whole procedure, intermediate values may be interpolated using a fitted response surface.

For the example of Balsa wood, the “optimal" damping factors and the corresponding smoothing factors resulting from a local Fourier analysis are shown in Fig. 16. Here, the angles η, φ, θ are spaced by $\frac{\pi}{30}$ and the damping factor ω is spaced by 0.01. Application of spatial Jacobi smoothing to the example problem 10a with constant orientations leads to the iteration number plots shown in Fig. 17. There, both designs (initial and topology optimized) are used in combination with four grids. Since the orientations are constant, the damping factor ω is also constant. However, it is adjusted to the occurring orientation instead of using the standard value $\omega = 0.25$. The spatial Jacobi method is stable for all orientations. Compared to a damping factor of $\omega = 0.25$, the number of iterations is reduced by up to 20%, depending on the specific orientation.

A similar improvement is obtained when considering orientations that are aligned with the principal stress direction. Table 5 shows the iteration numbers for both designs and both cycles for four grids. Here, actually spatial damping factors are practically used with values $\omega = 0.25...0.45$. In all situations, the number of iterations is reduced by more than 15%. Considering spatially random orientations instead, the improvement is still 7–10%.

Using spatial Jacobi smoothing instead of classic Jacobi smoothing, the computation time per iteration increases less than 1%. Additionally, the lookup table must be computed once. Using a workstation, this takes approximately 10 min, independent of the size of the topology optimization problem.

6 Recommendations to choose smoother for different situations

Combining the analysis of the computational effort with the obtained numbers of iterations shows that plane smoothing and alternating plane smoothing usually do not improve the overall computation time. The reduction in solver iteration is fully covered by a significant increase in cost for a single smoothing swipe.

The newly presented spatial Jacobi smoothing always outperforms standard Jacobi smoothing since it reduces the iterations without increasing the cost. Hence, it should always be preferred.

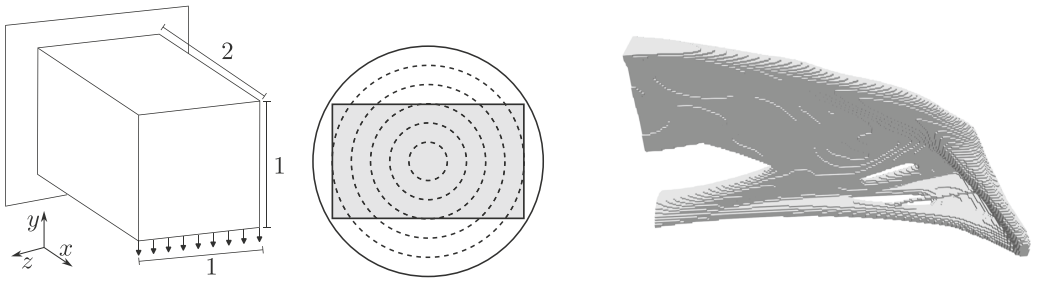
For the remaining methods, the situation is less obvious. Lexicographic SOR reduces the number of iterations by approximately 50–70% compared to spatial Jacobi, but it increases the cost per smoothing swipe. Still, it might outperform Jacobi and spatial Jacobi by 10–20% considering the number of floating-point multiplications. However, there is a huge amount of indexing leading to higher observed computation times in Matlab. Additionally, it is not parallelizable. In the authors opinion, the possible advantages are too low to switch to lexicographic SOR.

If the material orientation varies over the design domain, coupled SOR and line SOR reduce the required number of iterations by up to 70% to similar iteration numbers. Additionally, both methods parallelize well. Since coupled SOR requires significantly fewer operations than line SOR, it should be preferred in these situations. Depending on the efficiency of the implementation, it might outperform the spatial Jacobi method, and hence, could be preferred.

If the orientation field is (nearly) constant, the performance depends on the specific orientation as well as the density field. In many cases, the density field reduces the performance differences such that spatial Jacobi smoothing shows the best performance. However, especially at orientations that do not align with the grid, coupled SOR outperforms all competitors. On the other hand, line SOR often outperforms all other methods if the lines align with the material orientation. For a good decision, the reader is advised to check the results of the local Fourier analysis.

7 Numeric optimizations

To validate the previous observations, several methods are used in a (material orientation and) topology optimization. The required number of multigrid iterations to reach a relative tolerance of 10^{-10} is tracked during the optimization process. For the topology optimization, the modified SIMP approach (Sigmund 2007) in combination with the PDE filter with consistent boundary conditions (Wallin et al. 2020) is used. To keep the examples as simple as possible, no continuation



(a) Design space, loads and boundary conditions for the cantilever topology optimization.

(b) Position of the design space in the Balsa tree it is cut off.

(c) Optimized design of the cantilever example. Shown is the iso-cut where all elements with a density that is higher than 0.5 are shown.

Fig. 18 Optimization problem and result of the cantilever beam example

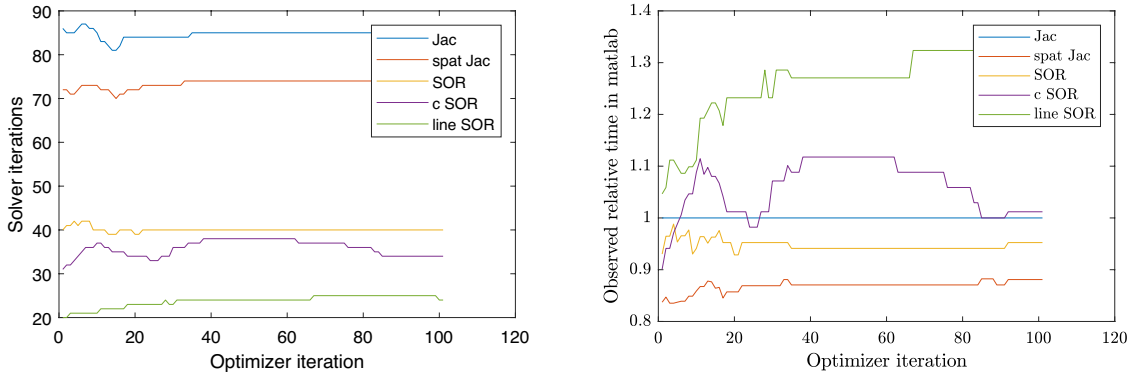


Fig. 19 Number of solver iterations (left) and observed computation time (right) during the optimization of the Balsa cantilever example for the following smoothing algorithms: Jacobi method ("Jac", $\omega = 0.25$, Sect. 3.1), spatial Jacobi method ("spat Jac", Sect. 5), lexicographic SOR ("SOR", Sect. 3.2), coupled multicolor SOR ("c SOR", Sect. 3.3) and line SOR (Sect. 3.4). The observed time is given relative to the Jacobi method

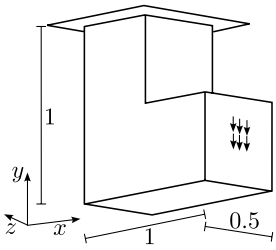
schemes are used. The optimizations are carried out using the method of moving asymptotes (Svanberg 1987) with the standard parameters. Only the initial move limit is set to 0.25 instead of 0.5. The optimizations are stopped after 100 iterations. In all cases considered, the relative change of the objective function is less than 10^{-4}.

7.1 Topology optimization of a Balsa wood cantilever beam

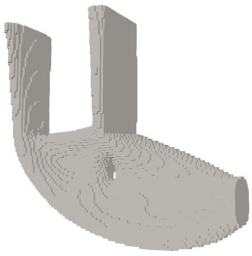
The first example consists of a simple compliance topology optimization of a cantilever beam. It is similar to the cantilever example in Aage et al. (2015). The optimization problem reads

$$\begin{aligned} &\min_{\boldsymbol{\rho}} f(\boldsymbol{\rho}) = \boldsymbol{u}^T \boldsymbol{K} \boldsymbol{u} \\ &\text{s.t.}: \quad V(\boldsymbol{\rho}) \leq 0.12 V_0 \\ &\qquad\quad 0 \leq \boldsymbol{\rho} \leq 1 \\ &\text{with}: \quad \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{u} = \boldsymbol{f} \end{aligned} , \tag{12}$$

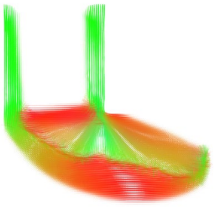
where $\boldsymbol{\rho}$ represents the vector of pseudo densities, $V(\boldsymbol{\rho})$ the volume at the current design and V_0 the design space volume. The design space, loads and boundaries are depicted in Fig. 18a . It is discretized using $192 \times 96 \times 96$ finite elements. The filter radius is set to 0.03, and the minimal Young's modulus is set to $E_{min} = 10^{-9} E_0$. The volume fraction is set to 12%. As material, Balsa wood with the material properties listed in Appendix D.2 is used. The spatially different material orientations are chosen such that the axial



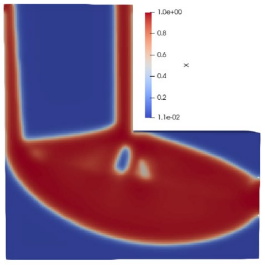
(a) Design space, loads and boundary conditions for the material orientation and topology optimization of the L-Beam example.



(b) Iso-cut at 0.5 of the density field of the optimized design of the L-beam example.



(c) Orientation field of the optimized design of the L-beam example. The colors highlight the orientations: green - aligned with y-axis, red - aligned with x-axis.



(d) Cut of the density field of the optimized design of the L-beam example. The cut is done at the model center.

Fig. 20 Optimization problem and result of the L-beam example

material axis points in the x-direction. To model imperfect growth and effects of branches, the axial axis is perturbed by random effects. The deviation in the axis is realized by perturbing the angles φ, θ using a Gaussian square exponential random field with the correlation length $l_c = 0.1$ and the variance $\sigma^2 = (18°)^2$. The radial and tangential axes are chosen to point in radial and tangential directions corresponding to the center of the design space, as depicted in Fig. 18b.

Overall, one optimization is done for each of the following smoothers using the W-Cycle multigrid preconditioned gradient solver with four grids: Jacobi method ($\omega = 0.25$), spatial Jacobi method, decoupled successive overrelaxation ($\omega = 1$), coupled successive overrelaxation ($\omega = 1$) and x-line successive overrelaxation ($\omega = 1$). The subproblems of the box smoother are solved using Jacobi preconditioned conjugate gradients with a tolerance of 10^{-3}. All optimizations converge to the exactly same design, which is depicted in Fig. 18c . Also, all numerical values are exactly the same.

The design is totally different from the result with isotropic material shown in Aage et al. (2015). The optimizer tries to exploit the anisotropic material behavior by reducing the average angle to the axial material direction. Due to the (asymmetric) random orientations, an asymmetric design is found.

Figure 19 shows the required number of solver iterations for every optimization step and all considered multigrid smoothers. Additionally, the figure shows the computation time relative to the Jacobi method in a naive sequential implementation (see Table 2). These results are consistent with previous findings. Spatial Jacobi reduces the required iteration number by 10–20% compared to Jacobi with $\omega = 0.25$. As expected, x-line successive overrelaxation shows the lowest number of iterations, while the point SOR methods are in between. Considering computation time, the spatial Jacobi method shows the best performance, whereas the line smoothing suffers from its high complexity. The absolute number of solver iterations stays relatively low because the W-Cycle is used and the topology aligns with the material orientation. This effect has already been observed in the previous analysis with orientations into principal stress direction.

7.2 Material orientation and topology optimization of a carbon fiber L-beam

The second example consists of a material orientation and topology optimization of a L-shaped beam. The design space is made of carbon fiber reinforced polymer with the material properties shown in D.1. The optimization problem reads

$$\begin{aligned} \min_{\boldsymbol{\rho},\boldsymbol{\Phi}} & f(\boldsymbol{\rho},\boldsymbol{\Phi}) = \boldsymbol{u}^T \boldsymbol{K} \boldsymbol{u} \\ \text{s.t.}: \quad & V(\boldsymbol{\rho}) \leq 0.15 V_0 \\ & 0 \leq \boldsymbol{\rho} \leq 1 \\ & -2\pi \leq \boldsymbol{\Phi} \leq 2\pi \\ \text{with}: \quad & \boldsymbol{K}(\boldsymbol{\rho},\boldsymbol{\Phi})\boldsymbol{u} = \boldsymbol{f} \end{aligned} \quad , \tag{13}$$

where $\boldsymbol{\rho}$ represents the vector of pseudo densities, $\boldsymbol{\Phi}$ the material angles η, φ, θ, $V(\boldsymbol{\rho})$ the volume at the current design, and V_0 the design space volume. The density field

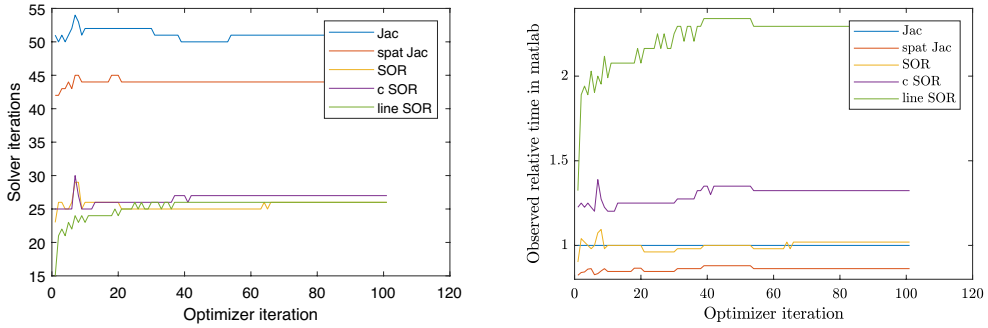


Fig. 21 Number of solver iterations (left) and observed computation time (right) during the optimization of the L-beam example for the following smoothing algorithms: Jacobi method ($\omega = 0.25$, Sect. 3.1), spatial Jacobi method (Sect. 5), lexicographic SOR (Sect. 3.2), coupled multicolor SOR (Sect. 3.3) and line SOR (Sect. 3.4). The observed time is given relative to the Jacobi method

is filtered using the PDE filter with consistent boundaries conditions and a filter radius of 0.03. The orientation field is filtered by a vector filter as described in Schmidt et al. (2020). There, the rotation angles are transformed to a direction vector. Afterwards, the direction vectors are filtered component-wise like the classic density filter in topology optimization. Again, the filter radius is set to 0.03. The problem is discretized using approximately 1,500,000 elements, and the volume fraction is set to 15%. As initial guess, all angles are set to 0°.

One full optimization is done for each of the following smoothers using the W-Cycle multigrid preconditioned gradient solver with four grids: Jacobi method ($\omega = 0.3$), spatial Jacobi method, decoupled successive overrelaxation ($\omega = 1$), coupled successive overrelaxation ($\omega = 1$) and x-line successive overrelaxation ($\omega = 1$). All optimizations lead to similar designs that are optically not distinguishable but have slightly different performance. Hence, the material orientation and topology optimization problem reacts very sensitive to small errors in the system solve. A iso-cut of the topology of the optimized design is shown in Fig. 20b , the orientation field in Fig. 20c. The general shape is similar to the results in literature, and the fibers are consequently aligned to the structural elements.

As in the cantilever example, the different smoothers lead to varying solution times. Figure 21 shows the required number of iterations to reach the desired tolerance of 10^{-10} for all optimization iterations. Additionally, the figure shows the computation time relative to the Jacobi method in a naive sequential implementation (see Table 2). As before, the standard Jacobi method shows the highest number of iterations. Using spatial Jacobi smoothing instead, the number of iterations drops by approximately 10–20% and leads to the lowest computation time. In contrast to the previous example, lexicographic SOR, coupled SOR and line SOR show similar iteration numbers. This is caused by the strongly varying material orientations and has also been observed in the previous analyses for Balsa. As shown in Fig. 20d , the final design shows clear contrasts and only contains intermediate densities at transition zones of the design surface. Hence, it is expected that the number of iterations grows as the design settles. In fact, the required number of iterations slightly increases during the first iterations. However, the final orientations are aligned with the structural elements, leading to the positive effects mentioned in Sect. 4.5.

8 Conclusions

In the current publication, the problem of simple Jacobi-smoothed multigrid methods with orthotropic materials is explained. Exemplary, Balsa wood and carbon fiber-reinforced polymers are considered. Different standard smoothers (namely the Jacobi method, lexicographic successive overrelaxation, multicolor coupled successive overrelaxation, multicolor line successive overrelaxation, multicolor plane successive overrelaxation and alternating multicolor plane successive overrelaxation) are analyzed using the local Fourier analysis and numeric experiments at different designs. There, constant material orientations as well as spatially varying orientations are considered. This way, the damping factor ω is chosen for the Jacobi method to be stable, and a new smoothing approach called spatial Jacobi smoothing is proposed. The new approach reduces the required number of solver iterations by up to 20% without increasing the cost per smoothing swipe.

When considering designs that have a developed topology, the problems of the multigrid method with orthotropic materials add up with the problems of contrast-rich designs. Depending on the material orientation field, the number of iterations totally explodes. In this case, using the

W-Cycle instead of the V-Cycle significantly improves the performance.

Based on the analyses, recommendations are given for choosing the smoothing algorithm for the multigrid method. For most more advanced smoothing algorithms, especially plane relaxation, the improved convergence properties do not balance the increased cost per smoothing swipe. Hence, the new approach is a good choice for many situations. However, depending on the implementation quality, multicolor coupled SOR and multicolor line SOR, might outperform the spatial Jacobi method.

Finally, the recommendations and results are validated at two different topology optimization examples.

Future research might address the generalization of the presented analyses and recommendations to the case of multi-material optimization, where also the type of material elastic symmetry varies element-wise.

Parallelization of multicolor SOR methods

All multicolor SOR methods use a chessboard-like node numbering (see Fig. 6b (b)), such that nodes of the same color have no couplings. On a structured 3D mesh, 8 colors are required. In the context of line smoothing, whole lines belong to the same color, such that the total number of numbers decreases to 4. Analogue, plane smoothing leads to 2 colors. In general, the state $\boldsymbol{u}_i$ can be rewritten as

$$\boldsymbol{u}_i = \begin{pmatrix} \boldsymbol{u}_{i,1} \\ \boldsymbol{u}_{i,2} \\ \vdots \\ \boldsymbol{u}_{i,n_c} \end{pmatrix}, \tag{A1}$$

where the partial vectors $\boldsymbol{u}_{i,c}$ correspond to the nodes of color c and n_c is the number of colors. Considering the coloring scheme, (7) is reformulated sequentially as

$$\boldsymbol{u}_{i+1,c} = \boldsymbol{u}_{i,c} + \left(\boldsymbol{L}_c + \frac{1}{\omega}\boldsymbol{D}_c\right)^{-1} \cdot \left(\boldsymbol{f}_c - \boldsymbol{K}_{c,:}\begin{pmatrix} \boldsymbol{u}_{i+1,1\ldots c-1} \\ \boldsymbol{u}_{i,c\ldots n_c} \end{pmatrix}\right), \tag{A2}$$

where $\boldsymbol{K}_{c,:}$ are the rows of the stiffness matrix $\boldsymbol{K}$ corresponding to color c, $\boldsymbol{u}_{i,1\ldots c-1}$ is the state vector corresponding to all previous colors and $\boldsymbol{u}_{i,c\ldots n_c}$ is the state vector of the current color and all following colors at iteration step i. The matrices $\boldsymbol{L}_c$ and $\boldsymbol{D}_c$ represent the lower diagonal and the diagonal part of the stiffness matrix entries corresponding to of color c. Due to the chosen coloring scheme, the matrix $\boldsymbol{L}_c$ only contains couplings between degrees of freedoms at the same node and hence, all nodes of the same color can be smoothed in parallel, whereas the different colors must be processed sequentially. In the case of coupled smoothing/line smoothing/plane smoothing, the matrix $\boldsymbol{L}_c$ vanishes and the matrix $\boldsymbol{D}_c$ becomes block diagonal. For more details, the reader is referred to Trottenberg et al. (2007).

Detailed analysis of computation time for different smoothing algorithms

The required computational cost of the different smoothing algorithms from Sect. 3.5 is determined based on the following assumptions and simplifications:

- Isometric 3D grid with first-order finite elements, standard grid transfer
- Cost of coarse grid solve is neglectable. This is true when using a sufficient number of grids. For example, five grids with approximately four billion elements on finest grid lead to approximately one million elements on the coarse grid. Then, a vector matrix product on the fine grid is more expensive than a (approximate) solve on the coarse grid.
- Perfect implementation such that computation time only depends on the number of multiplications
- The element stiffness matrices are already computed and stored in the memory
- The number of nodes n_N is approximately equal to number of elements

A major part of the multigrid method is to compute the vector matrix product $\boldsymbol{Ku}$. The (assembled) global stiffness matrix is extremely sparse. On an isometric 3D grid, each node is connected to 27 nodes, which have three degrees of freedom. Hence, each row of the stiffness matrix contains a maximum of 81 entries. Considering three rows per node, the cost of a vector matrix product is equal to $81 \cdot 3n_N = 243n_N$ multiplications. If the assembled stiffness matrix is not available, the vector matrix product is computed on element level. The element stiffness matrix is full and has 576 entries. Hence, the number of scalar multiplications is approximately $576n_N$. However, the procedure is parallelizable.

The Jacobi method only requires computing the residual (vector matrix product) and inverting a diagonal matrix. Hence, the additional cost is $3nN$ multiplications.

In the lexicographic successive overrelaxation, a lower diagonal matrix is inverted. If the lower diagonal matrix $\boldsymbol{L}$ is available, solving the system of equations is trivial, leading to $41 \cdot 3n_N = 123n_N$ additional multiplications. However, this framework is strictly sequential and requires some indexing, leading to higher computational cost. If $\boldsymbol{L}$ is not available, the system of equations must be solved on element level. This means, sequentially for every node, the couplings to all neighbor nodes must be computed from the

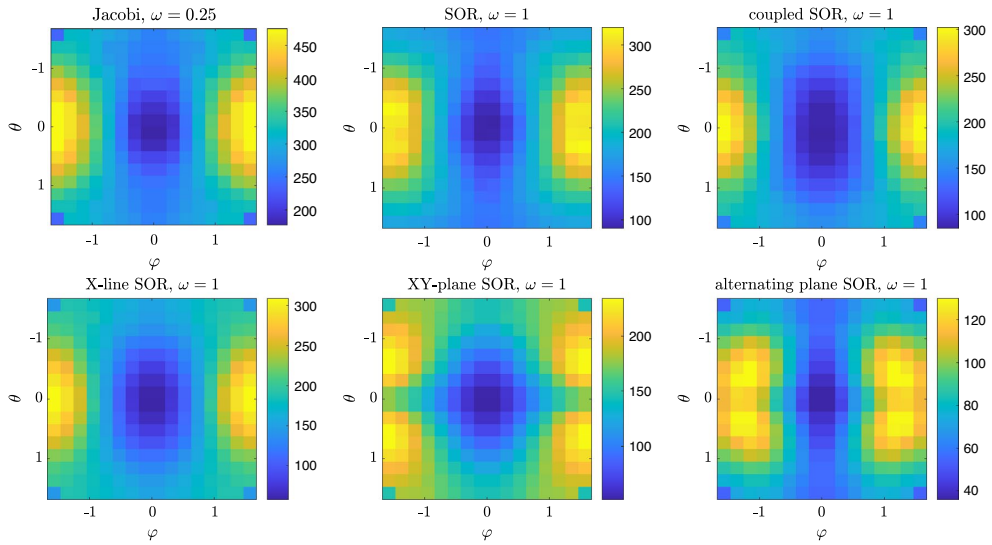


Fig. 22 Required number of iterations for the multigrid method to reach a relative tolerance of 10^{-10} for different constant material orientations φ, θ of Balsa wood. Here, the elasticity equations of the example problem 10b with optimized densities is solved using 4 grids. The colors indicate the required number of iterations

corresponding element stiffness matrices. The couplings are afterwards multiplied with the corresponding entries of the deformation vectors, leading to approximately $576n_N$ additional multiplications and a huge amount of indexing. The overall procedure again is fully sequential.

The remaining methods are based on a multicolor pattern and the inversion of a block diagonal matrix. In every sequence, the residual is computed for all degrees of freedom of the same color. Afterwards, the block-diagonal system of equations $\boldsymbol{D}_c^{-1}\left(\boldsymbol{f_c} - \boldsymbol{K}_{\boldsymbol{c},:}\begin{pmatrix}\boldsymbol{u}_{i+1,1,1\ldots c-1}\\ \boldsymbol{u}_{i,c\ldots n_c}\end{pmatrix}\right)$ is solved, leading to a state update. Using the updated state, the next sequence starts. Solving the system of equations with conjugate gradients and a tolerance of 10^{-2} requires approximately 5–10 iterations for the coupled SOR and 10–20 iterations for the line and plane smoothers. Each degree of freedom has contact with 3 (pointwise), 9 (line) or 27 (plane) other degrees of freedom, leading to a computational cost of 9 (pointwise), 27 (line) or 81 (plane) scalar multiplications per node and inner iteration. The overall framework is simply parallelized since the content of the sequences is parallelizable.

Besides the theoretical computational effort, data indexing, copying and other operations increase the computation time. However, highly efficient implementations might reach the theoretical results.

Iteration plots for topology optimized design and four grids

See Fig. 22.

Material properties of different orthotropic materials

In the following, typical material properties of different orthotropic materials are collected or computed.

Carbon fiber reinforced composites

For zero rotations and Voigt notation, the material compliance tensor $\boldsymbol{s}$ of transversal isotropic material reads

$$\boldsymbol{s} = \begin{pmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{12}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{\nu_{23}}{E_2} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{12}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{23}} \end{pmatrix}. \tag{D3}$$

Typical values, which are used in the paper, are $E_1 = 140GPa$, $E_2 = 12GPa$, $G_{12} = 5.8GPa$, $G_{23} = 5.4GPa$, $\nu_{12} = 0.26$ and $\nu_{23} = 0.11$.

Balsa wood

The mechanical properties of Balsa wood are taken from Wang et al. (2019). For zero rotations and Nye notation, the material compliance tensor s reads

$$s = \begin{pmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & \frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{12}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{23}} \end{pmatrix} \tag{D4}$$

with $E_1 = 6.3GPa$, $E_2 = 0.11GPa$, $E_3 = 0.3GPa$, $G_{12} = 0.2GPa, G_{13} = 0.31GPa, G_{23} = 0.03GPa, \nu_{12} = 0.49$, $\nu_{23} = 0.24$ and $\nu_{13} = 0.23$.

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

Author contributions Jan Krüger: Conceptualization, Methodology, Software, Investigation, Writing original draft. Benedikt Kriegesmann: Review and Editing, Funding acquisition, Supervision.

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. The underlying code will be made available at request.

References

Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: an easy-to-use, fully parallel, open source topology optimization framework. Struct Multidisc Optim 51(3):565–572. https://doi.org/10.1007/s00158-014-1157-0

Aage N, Andreassen E, Lazarov BS, Sigmund O (2017) Giga-voxel computational morphogenesis for structural design. Nature 550(7674):84–86. https://doi.org/10.1038/nature23911

Amir O, Bendsøe MP, Sigmund O (2009) Approximate reanalysis in topology optimization. Int J Numer Meth Eng 78(12):1474–1491. https://doi.org/10.1002/nme.2536

Amir O, Sigmund O (2011) On reducing computational effort in topology optimization: how far can we go? Struct Multidisc Optim 44(1):25–29. https://doi.org/10.1007/s00158-010-0586-7

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

Arms RJ, Gates LD, Zondek B (1956) A method of block iteration. J Soc Ind Appl Math 4(4):220–229. https://doi.org/10.1137/0104012

Borrvall T, Petersson J (2001) Large-scale topology optimization in 3D using parallel computing. Comput Methods Appl Mech Eng 190(46):6201–6229. https://doi.org/10.1016/S0045-7825(01)00216-X

Briggs WL, Henson VE, McCormick SF (2000) A multigrid tutorial, 2nd edn. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719505

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

Da Silva A, Kyriakides S (2007) Compressive response and failure of balsa wood. Int J Solids Struct 44(25):8685–8717. https://doi.org/10.1016/j.ijsolstr.2007.07.003

Davis TA (2006) Direct methods for sparse linear systems: fundamentals of algorithms. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718881

Dreyer T, Maar B, Schulz V (2000) Multigrid optimization in applications. J Comput Appl Math 120(1):67–84. https://doi.org/10.1016/S0377-0427(00)00304-6

Evans DJ, Biggins MJ (1982) The solution of elliptic partial differential equations by a new block over-relaxation technique. Int J Comput Math 10(3–4):269–282. https://doi.org/10.1080/00207168208803287

Evgrafov A, Rupp CJ, Maute K, Dunn ML (2008) Large-scale parallel topology optimization using a dual-primal substructuring solver. Struct Multidisc Optim 36(4):329–345. https://doi.org/10.1007/s00158-007-0190-7

Ferguson WE Jr (1986) The rate of convergence of a class of block Jacobi schemes. SIAM J Numer Anal 23(2):297–303. https://doi.org/10.1137/0723021

Fulton SR, Ciesielski PE, Schubert WH (1986) Multigrid methods for elliptic problems: a review. Mon Weather Rev 114(5):943–959

Hammer VB, Olhoff N (2000) Topology optimization of continuum structures subjected to pressure loading. Struct Multidisc Optim 19(2):85–92. https://doi.org/10.1007/s001580050088

Herrero-Pérez D, Martínez Castejón PJ (2021) Multi-GPU acceleration of large-scale density-based topology optimization. Adv Eng Softw 157–158:103006. https://doi.org/10.1016/j.advengsoft.2021.103006

Hestenes MR, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. J Res Natl Bur Stand 49(6):2379

Jantos DR, Hackl K, Junker P (2020) Topology optimization with anisotropic materials, including a filter to smooth fiber pathways. Struct Multidisc Optim 61(5):2135–2154. https://doi.org/10.1007/s00158-019-02461-x. (**Accessed 2020-06-28**)

Kipping J, Schüppstuhl T (2023) Load-oriented nonplanar additive manufacturing method for optimized continuous carbon fiber parts. Materials 16(3):998. https://doi.org/10.3390/ma16030998

Lazarov BS, Sigmund O (2011) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Meth Eng 86(6):765–781. https://doi.org/10.1002/nme.3072
Lee J, Kim D, Nomura T, Dede EM, Yoo J (2018) Topology optimization for continuous and discrete orientation design of functionally graded fiber-reinforced composite structures. Compos Struct 201:217–233. https://doi.org/10.1016/j.compstruct.2018.06.020
Liu H, Hu Y, Zhu B, Matusik W, Sifakis E (2018) Narrow-band topology optimization on a sparsely populated grid. ACM Trans Graph 37(6):251–125114. https://doi.org/10.1145/3272127.3275012
Lund E (2017) Discrete material and thickness optimization of laminated composite structures including failure criteria. Struct Multidisc Optim. https://doi.org/10.1007/s00158-017-1866-2
Luo S, Yang F, Wang Y (2024) An efficient isogeometric topology optimization based on the adaptive damped geometric multigrid method. Adv Eng Softw 196:103712. https://doi.org/10.1016/j.advengsoft.2024.103712
Ma J, Li Z, Zhao Z-L, Xie YM (2021) Creating novel furniture through topology optimization and advanced manufacturing. Rapid Prototyp J 27(9):1749–1758. https://doi.org/10.1108/RPJ-03-2021-0047
Mahdavi A, Balaji R, Frecker M, Mockensturm EM (2006) Topology optimization of 2D continua for minimum compliance using parallel computing. Struct Multidisc Optim 32(2):121–132. https://doi.org/10.1007/s00158-006-0006-1
Montemurro M, Mas A, Zerrouq S-E (2024) Topology and anisotropy optimisation of continua using non-uniform rational basis spline entities. Comput Methods Appl Mech Eng 420:116714. https://doi.org/10.1016/j.cma.2023.116714
Morano E, Mavriplis DJ, Venkatakrishnan V (1998) Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems. SIAM J Sci Comput 20(2):393–415. https://doi.org/10.1137/S1064827595287638
Oosterlee CW (1997) A GMRES-based plane smoother in multigrid to solve 3D anisotropic fluid flow problems. J Comput Phys 130(1):41–53. https://doi.org/10.1006/jcph.1996.5442
Padhi AP, Chakraborty S, Chakrabarti A, Chowdhury R (2023) Efficient hybrid topology optimization using GPU and homogenization-based multigrid approach. Eng Comput 39(5):3593–3615. https://doi.org/10.1007/s00366-022-01771-x
Peetz D, Elbanna A (2021) On the use of multigrid preconditioners for topology optimization. Struct Multidisc Optim 63(2):835–853. https://doi.org/10.1007/s00158-020-02750-w
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
Schmidt M-P, Couret L, Gout C, Pedersen CBW (2020) Structural topology optimization with smoothly varying fiber orientations. Struct Multidisc Optim 62(6):3105–3126. https://doi.org/10.1007/s00158-020-02657-6
Shaidurov VV (1995) Multigrid methods for finite elements. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-8527-9
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
Steltner K, Kipping J, Kriegesmann B, Schüppstuhl T (2025) A workflow for designing stiffness-optimized structures in the context of additive manufacturing of endless fiber-reinforced composites, Journal of Thermoplast Composite Materials. https://doi.org/10.1177/08927057251332788
Stüben K, Trottenberg U (1982) Multigrid methods: fundamental algorithms, model problem analysis and applications. In: Hackbusch W, Trottenberg U (eds) Multigrid methods. Lecture Notes in Mathematics. Springer, Berlin, pp 1–176. https://doi.org/10.1007/BFb0069928
Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Meth Eng 24(2):359–373. https://doi.org/10.1002/nme.1620240207
Thibaut B, Gril J, Fournier M (2001) Mechanics of wood and trees: some new highlights for an old story. Comptes Rendus de l'Académie des Sciences—Series IIB—Mechanics 329(9):701–716. https://doi.org/10.1016/S1620-7742(01)01380-0
Thole C-A, Trottenberg U (1986) Basic smoothing procedures for the multigrid treatment of elliptic 3D operators. Appl Math Comput 19(1):333–345. https://doi.org/10.1016/0096-3003(86)90112-8
Träff EA, Sigmund O, Aage N (2021) Topology optimization of ultra high resolution shell structures. Thin-Walled Struct 160:107349. https://doi.org/10.1016/j.tws.2020.107349
Trottenberg U, Oosterlee CW, Schüller A, Brandt A, Oswald P, Stüben K (2007) Multigrid. Transferred to digital print edn. Elsevier Academic Press, Amsterdam Heidelberg
Vito AF, Vicente WM, Xie YM (2023) Topology optimization applied to the core of structural engineered wood product. Structures 48:1567–1575. https://doi.org/10.1016/j.istruc.2023.01.036
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 S, Sturler ED, Paulino GH (2007) Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. Int J Numer Meth Eng 69(12):2441–2468. https://doi.org/10.1002/nme.1798
Wang Z, Wang Y, Cao Y, Gao Z (2019) Measurements of the shear modulus of materials by the free-plate torsional mode shape method. J Test Eval 47(2):1163–1181. https://doi.org/10.1520/JTE20160471
Wesseling P (1992) An introduction to multigrid methods. John Wiley & Sons Ltd, Chichester
Wienands R, Joppich W (2004) Practical Fourier analysis for multigrid methods. Chapman and Hall/CRC, New York. https://doi.org/10.1201/9781420034998
Wu J, Dick C, Westermann R (2016) A system for high-resolution topology optimization. IEEE Trans Visual Comput Graphics 22(3):1195–1208. https://doi.org/10.1109/TVCG.2015.2502588

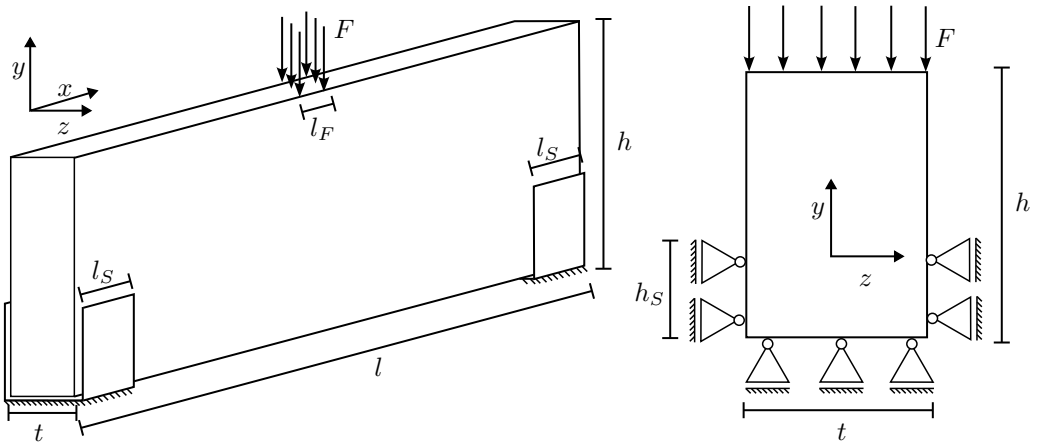


Figure 5.1: Model sketch of the wooden three-point bending beam. On the left, a parametric view is shown, on the right, the cut of the structure is depicted.

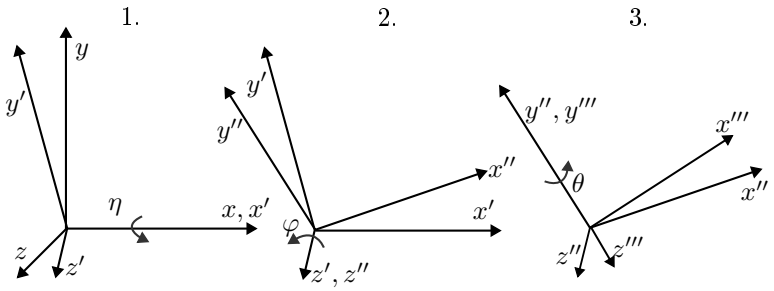
dimensions	support domain	load F	load domain	No. elements
$490 \times 90 \times 8\text{mm}^3$	$l_S = 20\text{mm}$, $h_S = 30mm$	100N	$l_F = 4\text{mm}$	$784 \times 288 \times 16$

Table 5.1: Model parameters of the wooden three-point bending beam example

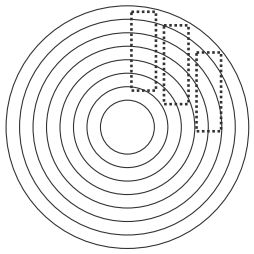
5.2 Optimization of a wooden three point bending beam under multiple uncertainties

Using the previously discussed probabilistic methods, as well as the modified multigrid method with the spatial Jacobian smoothing approach, a robust design optimization is applied to a modified three-point bending beam. A sketch of the model is depicted in figure 5.1, the basic model parameters are collected in table 5.1. The problem consists of a rectangular design space with the length $l = 490\text{mm}$, the height $h = 90\text{mm}$ and the thickness 5mm, which is discretized using $784 \times 288 \times 16$ hexagonal finite elements with the aspect ratio $2 \times 1 \times 1$. A rather high aspect ratio is chosen to support the convergence of the multigrid solver, similar to semicoarsening [57]. The model is loaded by a surface load $F = 1000\text{N}$ which is distributed over the whole thickness and the length $l_F = 4\text{mm}$ at the center of the top surface.

In difference to the classic three-point bending beam benchmark example, the structure is not supported by a rotational support at the outer lower edges. Instead, the structure is supported in y-direction on the lower surface (length $l_S = 20\text{mm}$) near the outer edges, which is chosen to model a structure laying on a surface. To prevent the structure from tilting into z-direction, additional supports in z-direction are added at the front and at the back surface (length $l_S = 20\text{mm}$, height $h_S = 30\text{mm}$) near the lower outer edges. Finally, a single loaded node is supported in x-direction, such that the resulting system of equations is not singular. However, it does not influence the physical outcome since no loads act in x-direction. The model is made of a spruce sheet, where the material orientation is defined by the spatial rotation angles η_i, φ_i and θ_i corresponding to element i. At zero rotation, the stiffest direction aligns with the x-axis and the lowest stiffness direction



(a) Rotations that represent the material orientation. The first rotation (angle η) acts around the initial x-axis, the second rotation (angle φ) acts around the rotated z'-axis and the final rotation (angle θ) acts around the rotated y''-axis.



(b) Different possible positions of the wooden sheet in the original wood trunk. The view corresponds to the $y-z$-plane in figure 5.1 and the lines represent the annual rings.

Figure 5.2: Material orientations of the wood.

aligns with the y-axis. The rotations are depicted in figure 5.2a. The first rotation (angle η) acts around the initial x-axis, the second rotation (angle φ) acts around the rotated z'-axis and the final rotation (angle θ) acts around the rotated y''-axis. Since the sheet is cut from an original trunk as depicted in figure 5.2b, the rotation angle η depends on the cutting position. Due to the small thickness of the sheet it is assumed to be constant in z-direction, however, depending on the cut position, it is not constant in y-direction. To reduce the complexity, it is assumed that the rotation angle $\eta(\boldsymbol{z})$ does not change in axial direction and depends linearly on the y-positions such that it reads

$$\eta(\boldsymbol{z}) = \eta_0 + \frac{\Delta\eta}{100\text{mm}} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \boldsymbol{z} \tag{5.1}$$

with the slope $\Delta\eta$ and the value at the center η_0.

The material properties are obtained experimentally by application of compression tests in fiber direction (see next section). All other material parameters are obtained from the measured principal stiffness using the relations presented in [153]. Additionally, it is assumed that the wood fibers are oriented in x-direction, i.e., $\varphi = 0, \theta = 0$ in the mean case. As mentioned before, the angle η is expected to be changing linearly in y-direction and the parameters η_0 and $\boldsymbol{\Delta}\eta$ are obtained experimentally.

0, 0 A	1, 0 A	2, 0 A	3, 0 A	4, 0 A
0, 1 A	1, 1 A	2, 1 A	3, 1 A	4, 1 A
0, 2 A	1, 2 A	2, 2 A	3, 2 A	4, 2 A
0, 3 A	1, 3 A	2, 3 A	3, 3 A	4, 3 A
0, 4 A	1, 4 A	2, 4 A	3, 4 A	4, 4 A

Figure 5.3: Generation of test specimens from a wooden sheet. The lines represent the cut geometry, the specimen number is engraved to assign each specimen to its position on the sheet and to the sheet number.

5.2.1 Quantification of uncertain material and geometry properties

For the robust optimization, the following uncertainties are considered:

- spatially varying principal stiffness $E_1(\boldsymbol{z})$
- spatially varying out-of plane geometric imperfections $\Delta\boldsymbol{z}(\boldsymbol{z})$
- spatially varying fiber orientations $\varphi(\boldsymbol{z})$ and $\theta(\boldsymbol{z})$
- global orientation parameters η_0 and $\boldsymbol{\Delta}\eta$

It is assumed that there is correlation between variables of the same type, but random variables of different types are expected to be independent of each other. For simplicity, all random variables are assumed to be normal distributed. All research data related to the experiments is published in non-modified form at [154].

Compression tests for determination of the principal Young's modulus

Although the mechanical properties (including stochastic distributions) of wood have been determined experimentally in several publications (see e.g. [155] for spruce), it seems like spatial stochastic properties have not been treated yet. However, these properties are to be considered in the following robust design optimization and hence, are measured in the following. The spatial principal Young's moduli are measured using compression tests. The specimens with the length $l = 50$mm, the height $h = 10$mm and the thickness $t = 5$mm are fabricated from a spruce sheet with the thickness $t = 5$mm, where the fibers are oriented into the longest direction. The final optimized results of the following chapter are fabricated from similar sheets of the same production batch. In order to obtain spatial stochastic properties (especially the covariance function), 19×9 specimens are fabricated from the same sheet and every specimen is uniquely numbered as depicted in figure 5.3 such that its original position can be recovered. In total 513 specimens have been fabricated and approximately 300 specimens have been tested, whereas the rest was destroyed due to user mistakes.

The testing setup is depicted in figure 5.4. The specimens are loaded by a compression testing machine of the type "Zwick/Roell Z010" with a force sensor of the type "Zwick/Roell Xforce FP". The deformation of the structure is obtained using the internal position sensor of the machine, under the assumption of constant strains. The force is applied directly by the machine without the

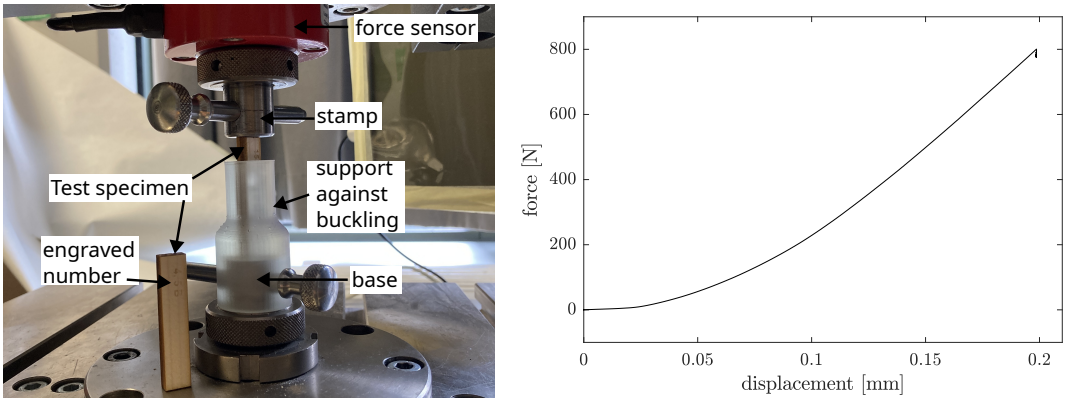


Figure 5.4: Left: Experimental setup for the determination of the (spatial) stochastic properties of the principal stiffness of spruce wood. Right: Typical load displacement curve of a compression test.

use of any additional clamps, to remove the effects of any connecting elements, like screws. Due to the high aspect ratio, the specimens are prone to buckling. Additionally, eccentric or even tilted positioning of the specimens might lead to false results due to stress concentrations. Therefore, a support preventing imperfect positioning and buckling is constructed and 3D-printed. In theory, the support might manipulate the result due to friction and transverse deformations. However, it is expected that this effect is rather low due to the low stiffness of the support and a lash between specimen and support.

By the experiments, load displacement curves as depicted in figure 5.4 are obtained. For small displacements below 0.1mm, the figure shows progressive behavior, which is caused by starting effects, such as a aligning of the specimen and getting full contact between specimen and stamp. Afterwards, the figure shows nearly linear behavior. Therefore, the principal Young's modulus is computed by considering the slope between a load of 500N and 750N. The machine stops at 800N, which occurs at 0.2mm in this example. Until this point, no plastic deformation is observed. Unexpectedly, the force decreases by approximately $1-2\%$ after the machine stops. The drop highly depends on the compression velocity, and hence, it is expected to be some viscoelastic behavior. Therefore, the author chose to use a low compression velocity of $\frac{0.01}{60}\frac{\text{mm}}{\text{s}}$, which leads to the depicted result. If the velocity is chosen at $\frac{0.1}{60}\frac{\text{mm}}{\text{s}}$ instead, the drop in the end is much higher with approximately 10%. Besides, deformation of the testing machine itself induces a systematic error. Therefore, the stiffness of the machine is determined by application of the experiment without a specimen. The obtained machine stiffness is considered in the post processing of the experimental data.

Application of the experiment leads to an expected principal stiffness of 7.9GPa, which is in the same range as values found in literature [153, 156]. Considering the spatial information, the semivariogram depicted in figure 5.5 is found. There, the semivariance is shown for different distances in x- and y-direction. At very low distances, already a semivariance of $5 \cdot 10^5 (\text{MPa})^2$ is found, which stays nearly constant in y-direction, but increases in x-direction. Considering the nugget to be $5 \cdot 10^5 (\text{MPa})^2$, the random field seems to be only dependent on the x-distance. Besides the full variogram, also the 1D variogram for $\Delta y = 0$ is depicted in figure 5.5. Using optimization methods,

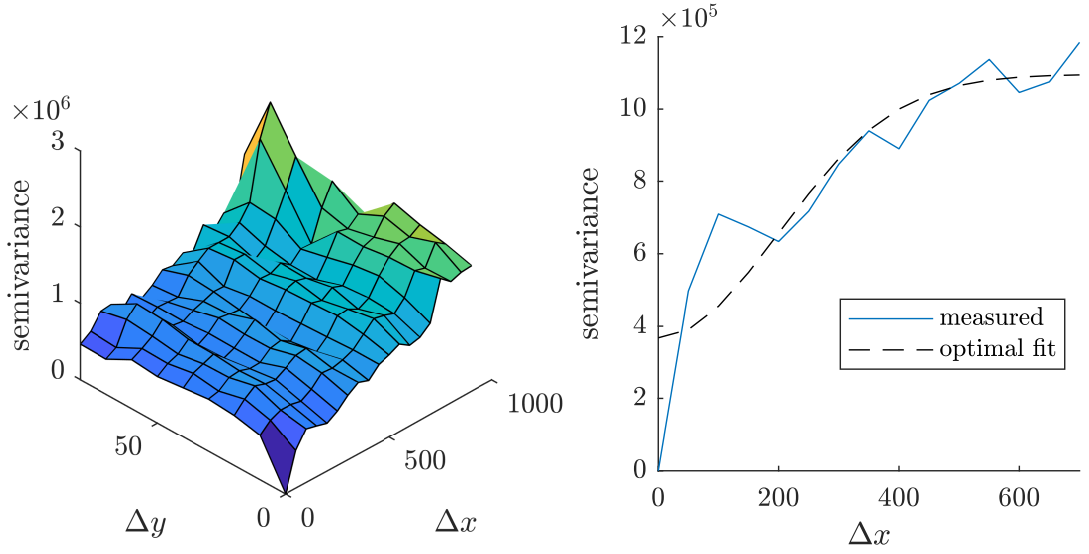


Figure 5.5: Left: Semivariogram of the measured principal Young's modulus. Right: Part of variogram where $\Delta y = 0$ and a fitted square exponential random field considering a nugget.

a square-exponential random field is fitted to the data and depicted by the dashed lines. From the authors point of view, the found random field shows acceptable alignment to the data.

In general, the nugget is caused by additional random effects of the experiment, or it represents random field components with a very low correlation length. Since the author put a high effort to reduce random effects of the experiments, it is expected that at least some part of the nugget is caused by components with a very low correlation length. A possible source of random components with a short correlation length is the annual ring structure of wood. Basically, the annual rings represent the hard wood which grew in winter, whereas the space in between represents the soft wood, which grew in the summer. By cutting the wood into the specimen, some specimens contain more annual rings than others, resulting in a different stiffness. The width of an annual ring is approximately 3mm. Considering this to be the major effect for the nugget, the overall covariance may be represented by a combination of random fields, where one random field only depends on the y-distance and has a correlation length of 3mm and the other random field only depends on the x-distance with a correlation length 300mm. Based on this interpretation of the experimental data, the covariance function is assumed to be

$$\operatorname{cov}(E_1(\boldsymbol{z}_i), E_2(\boldsymbol{z}_j)) = (705\text{MPa})^2 \cdot \exp\left(-\frac{(z_{1,i} - z_{1,j})^2}{(300\text{mm})^2}\right) + (780\text{MPa})^2 \cdot \exp\left(-\frac{(z_{2,i} - z_{2,j})^2}{(3\text{mm})^2}\right) \tag{5.2}$$

with $\boldsymbol{z}_i = \begin{pmatrix} z_{1,i} & z_{2,i} & z_{3,i} \end{pmatrix}^T$ for the following optimization and evaluation.

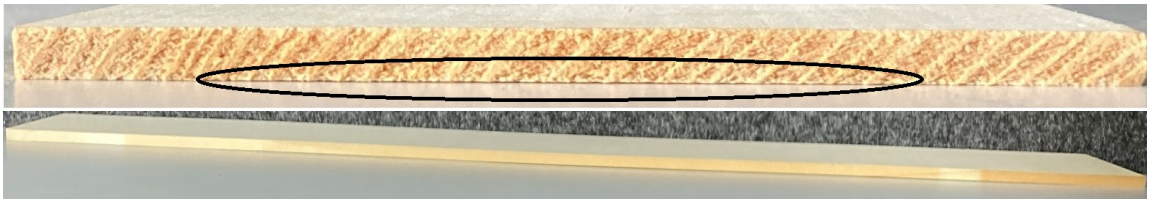


Figure 5.6: Out-of-plane imperfections of the wooden sheet in x-direction (top) and y-direction (bottom). The wooden sheet lies on an even plate. The tiny distance (marked in the top figure) between sheet and plate represents out-of-plane imperfections. The variance is obtained by inaccurate measurements of the imperfections at the highest amplitude.

Estimation of out-of-plane imperfections

Besides material uncertainties, geometric imperfections in out-of-plane direction influence the results. While in-plane imperfections are rather unlikely due to the high accuracy of the process of laser cutting, out-of-plane imperfections are caused by drying effects of the wood. Due to the structure of the annual rings, drying of the wood leads to a bending of the wooden sheet, as depicted in figure 5.6. To quantify the imperfections, it is assumed that the correlation is defined by an anisotropic random field, where the correlation lengths equal half of the original sheet dimensions, i.e. 500mm in x-direction and 50mm in y-direction. This is motivated by the observation that the imperfections have an arc like shape, where the imperfection is zero at the corners and maximal in the middle (or opposite). The variance is roughly approximated by inaccurate manual measurements as $\sigma_z^2 = (0.2\text{mm})^2$ in x-direction and $\sigma_z^2 = (0.1\text{mm})^2$ in y-direction. Due to the supports of the optimized model, the random field is conditioned such that the imperfections are zero at the supports. Finally, the used (non-conditioned) covariance function reads

$$\operatorname{cov}(\Delta z(\boldsymbol{z}_i), \Delta z(\boldsymbol{z}_j)) = (0.2\text{mm})^2 \cdot \exp\left(-\frac{(z_{1,i} - z_{1,j})^2}{(500\text{mm})^2}\right) + (0.1\text{mm})^2 \cdot \exp\left(-\frac{(z_{2,i} - z_{2,j})^2}{(50\text{mm})^2}\right) \quad (5.3)$$

with $\boldsymbol{z}_i = \begin{pmatrix} z_{1,i} & z_{2,i} & z_{3,i} \end{pmatrix}^T$.

Image processing for determination of material orientation

The material orientation is obtained by interpretation of the annual rings at the wooden sheet. It is assumed that the orientations of the annual rings in the $y-z$-plane correspond to the tangential material direction (i.e., lowest stiffness), whereas the axial material direction (i.e., highest stiffness) is obtained from the annual rings in the $x-y$-plane.

The determination of the stochastic properties of the material orientation angle η (i.e. the tangential material direction) is based on the framework summarized in figure 5.7. First, gray-scale images of the cut surface of the wooden sheet are generated using a conventional scanner. Secondly, the annual rings are marked manually at the outer positions for every specimen. For each marked line, the angle is computed using trigonometric relations. Since the spatial angle $\eta(\boldsymbol{z})$ is chosen to be defined by the parameters η_0 and $\Delta\eta$ with (5.1), these parameters are computed from the two

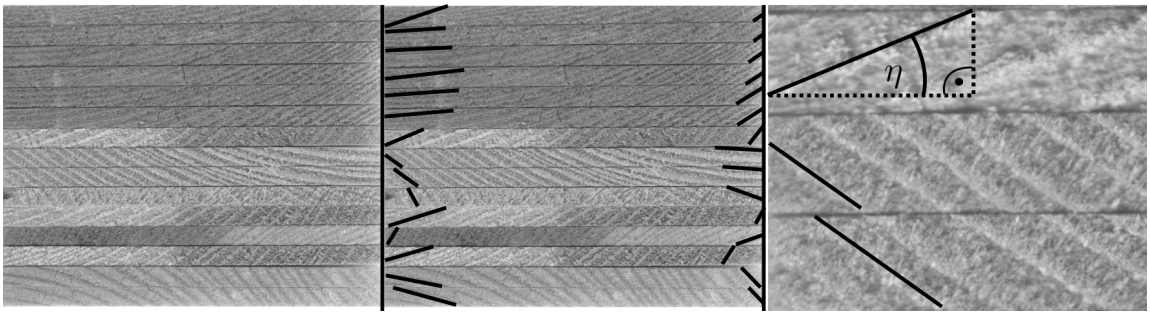


Figure 5.7: Framework of the determination of the material rotation angle $\eta(\boldsymbol{z})$ using manual image processing. The $y-z$-surface of multiple sheets is scanned (left), the annual rings are marked manually (middle) and the angles are computed using geometric relations (right).

angles obtained for each specimen. Hence, the parameters $\eta_{0,i}$ and $\Delta\eta_i$ of specimen i read

$$\eta_{0,i} = \frac{1}{2}\eta_i(\boldsymbol{z} = \begin{pmatrix}0 & -50\text{mm} & 0\end{pmatrix}^T) + \frac{1}{2}\eta_i(\boldsymbol{z} = \begin{pmatrix}0 & 50\text{mm} & 0\end{pmatrix}^T) \tag{5.4}$$

$$\Delta\eta_i = \eta_i(\boldsymbol{z} = \begin{pmatrix}0 & -50\text{mm} & 0\end{pmatrix}^T) - \eta_i(\boldsymbol{z} = \begin{pmatrix}0 & 50\text{mm} & 0\end{pmatrix}^T) \quad . \tag{5.5}$$

Considering 21 specimens, the mean values read $\mu_{\eta_0} = 50°$ and $\mu_{\Delta\eta} = \frac{23.5°}{100\text{mm}}$ and the covariance matrix reads

$$\boldsymbol{\Sigma}_\eta = \begin{pmatrix}(14°)^2 & 86°^2 \\ 86°^2 & (10°)^2\end{pmatrix} \quad . \tag{5.6}$$

The random properties of the material orientation angle φ are obtained by considering the annual rings in the $x-y$-plane. For accurate results it is required that the $x-y$-plane corresponds to the axial-radial-plane of the original tree. Therefore, only sheets are considered, where the annual rings are nearly orthogonal to the surface in the $y-z$-plane.

The general framework for the determination of spatial material angles $\varphi(\boldsymbol{z})$ is similar to the determination of $\eta(\boldsymbol{z})$. First, tangents of the annual rings are marked using lines and second, the spatial orientations are obtained using trigonometric relations to the lines. However, here, the lines are drawn automatically. The general framework is summarized in figure 5.8. First, the $x-y$-plane is scanned and the picture is clustered into several equidistant vertical slices. For every slice, the grey-scale information depending on the vertical coordinate is considered (and plotted). Each local minimum of this slice-wise function relates to a point on an annual ring. Application of the slicing and drawing the minimal points into the scanned figure leads to a set of points, which are placed on annual rings. Afterwards, the points on the same annual rings are connected to lines. As visible in figure 5.8, the framework falsely assigns some points to be on the annual rings. However, these points are not considered when drawing the lines. Finally, the orientations are obtained as the tangents of the lines.

The whole process is affected by noise on the image data leading to inaccurate the results. To increase the accuracy, filters are applied to the image data.

Application of the previously described image processing leads to the semivariogram depicted in figure 5.9. There, the semivariance is plotted for different x- and y-coordinates. Assuming that

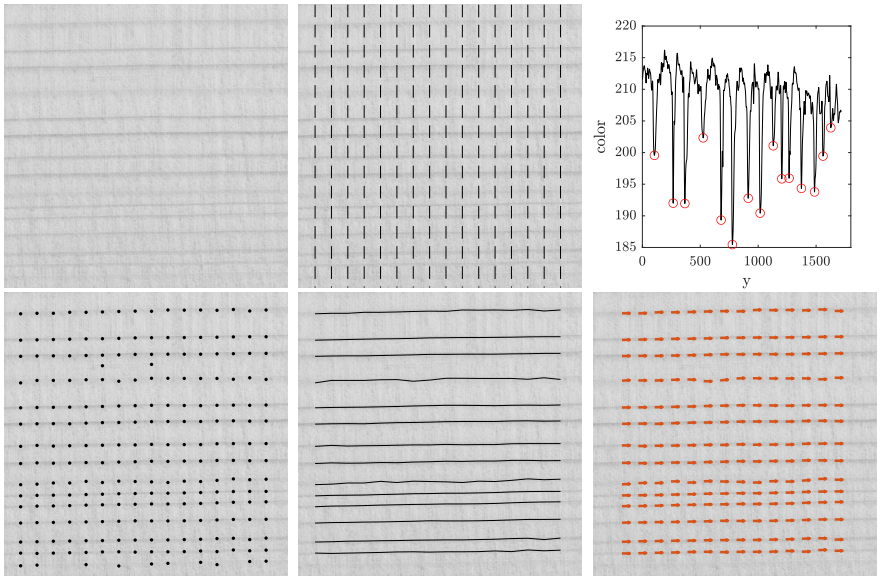


Figure 5.8: Framework of the determination of the material rotation angle $\varphi(\boldsymbol{z})$. The sheet is scanned (top left) and cut into slices (top middle). For each slice, the color value is plotted and the local minima are marked (top right). The minima are transferred to the scanned figure (bottom left) and connection of the points (bottom middle) leads to the material orientations (bottom right).

the total random field depends on two partial random fields, the covariance function

$$\operatorname{cov}(\varphi(\boldsymbol{z}_i), \varphi(\boldsymbol{z}_j)) = 0.6^{\circ^2} \cdot \exp\left[\left(\frac{(z_{1,i} - z_{1,j})}{6\text{mm}} + \frac{(z_{2,i} - z_{2,j})}{20\text{mm}}\right)^2\right] + 0.4^{\circ^2} \cdot \exp\left[\left(\frac{(z_{1,i} - z_{1,j})}{35\text{mm}}\right)^2\right] \tag{5.7}$$

with with $\boldsymbol{z}_i = \begin{pmatrix} z_{1,i} & z_{2,i} & z_{3,i} \end{pmatrix}^T$ accurately represents most data (see right figure 5.9.) and hence, is considered for the following robust design optimization.

The material rotation angles θ cannot be determined using the paradigm based on tracking the annual rings. However, it is assumed that the angle θ has the same stochastic properties as φ, but the angles are uncorrelated from each other. This assumption is motivated from considering the reasons for these imperfections. Both, imperfect angles φ and θ might be caused by bending of the tree, torsion of the tree and effects of far knots. Depending on the position, the effects are the same for both angles. For example, perfect bending of a perfect round tree leads to $\varphi(\boldsymbol{z}_1) = c$ and $\theta(\boldsymbol{z}_1) = 0$ at material point $\boldsymbol{z}_1$, but leads to $\varphi(\boldsymbol{z}_2) = 0$ and $\theta(\boldsymbol{z}_2) = c$ at material point $\boldsymbol{z}_2$, where c is some fixed value. In consequence, the covariance function

$$\operatorname{cov}(\theta(\boldsymbol{z}_i), \theta(\boldsymbol{z}_j)) = 0.6^{\circ^2} \cdot \exp\left[\left(\frac{z_{1,i} - z_{1,j}}{6\text{mm}} + \frac{(z_{2,i} - z_{2,j})}{20\text{mm}}\right)^2\right] + 0.4^{\circ^2} \cdot \exp\left[\left(\frac{z_{1,i} - z_{1,j}}{35\text{mm}}\right)^2\right] \tag{5.8}$$

with with $\boldsymbol{z}_i = \begin{pmatrix} z_{1,i} & z_{2,i} & z_{3,i} \end{pmatrix}^T$ is chosen for the material rotation angles θ.

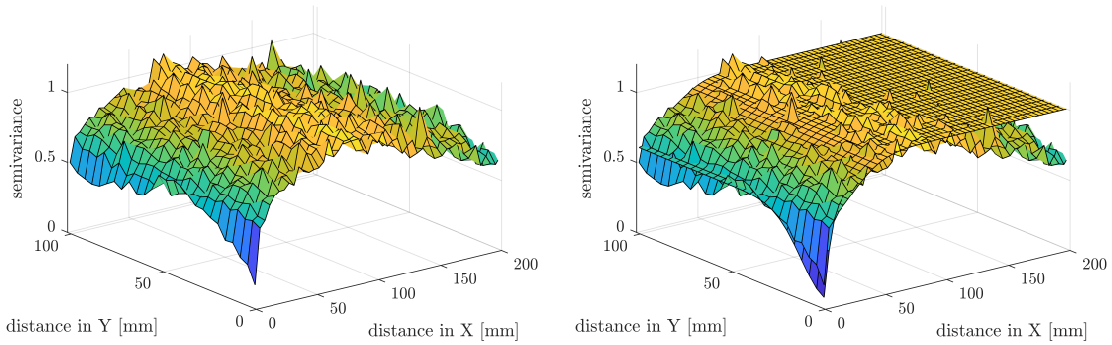


Figure 5.9: Semivariogram of the material rotation angle φ. Left, the original semivariogram is shown; right, the semivariogram of a fitted correlation function is plotted over the original variogram.

5.2.2 Numerical optimization

Using the design problem presented in figure 5.1 in combination with the (stochastic) material and geometry parameters obtained in 5.2.1, a compliance topology optimization considering volume constraints is performed. The deterministic optimization problem reads

$$\begin{aligned} &\min_{\boldsymbol{\rho}} c(\boldsymbol{\rho}) = \boldsymbol{F}^T \boldsymbol{u}(\boldsymbol{\rho}) \\ &\text{s.t.} \\ &0 \leq \rho_i \leq 1 \quad \forall i \\ &\frac{V(\boldsymbol{y}) - 0.3V_0}{0.3V_0} \leq \mathbf{0} \end{aligned} \quad , \tag{5.9}$$

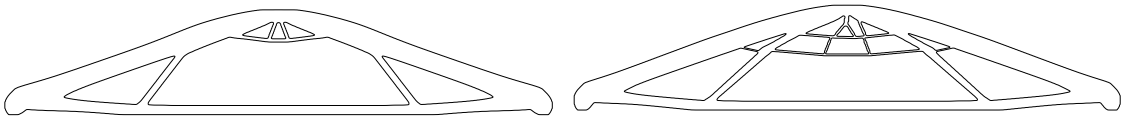
where V_0 represents the volume of the whole design space, $\boldsymbol{F}$ the load vector and $\boldsymbol{u}$ the displacement field. To prevent checkerboarding and enforce a minimal length scale the Helmholtz filter presented in [17] using consistent boundary conditions [19] with a equivalent filter radius of 1mm is applied. The material is modeled using the modified solid isotropic material with penalization (SIMP) scheme, although an anisotropic material is considered. Therefore, the material matrix $\mathcal{C}_i$ of element i is computed by

$$\mathcal{C}_i = 10^{-9}\mathcal{C}_0 + \rho_i^p(1 - 10^{-9})\mathcal{C}_0 \quad , \tag{5.10}$$

where $\mathcal{C}_0$ is the material matrix of solid material and the penalization parameter p is set to $p = 3$. Projection methods or continuation schemes are not applied. Although the problem is defined as a 3-dimensional problem, it should be fabricated using a laser cutting machine. Therefore, the problem is re-parameterized such that only two-dimensional topologies evolve by defining

$$\rho(x_i, y_i, z_j) = \hat{\rho}(x_i, y_i) \quad \forall j \tag{5.11}$$

with the new design densities $\hat{\boldsymbol{\rho}}$. Note that the filter (with zero padding) still acts in all three dimensions such that the edges are rounded up in all dimensions and the effective thickness is reduced to approximately 4.5mm. The optimization is done using the method of moving asymptotes



(a) Deterministic optimized. (b) Robust optimized using FOSM.

Figure 5.10: Optimized designs of the three-point bending beam example.

	deterministic compliance	probabilistic evaluation approach FOSM		Monte Carlo	
optimization approach	c	μ_c	σ_c	μ_c	σ_c
determ. optimization	88.49Nmm	88.49Nmm	9.046Nmm	87.37Nmm	9.498Nmm
RDO / FOSM	88.13Nmm	88.13Nmm	8.831Nmm	87.01Nmm	9.348Nmm

Table 5.2: Numerical results of the optimized three-point bending beam evaluated deterministically, using the first-order second-moment method (FOSM) and the Monte Carlo method with a sample size of 2500.

with the standard parameters presented in [12]. However, the initial asymptotes parameter ("sini") is set to 0.1 for a less aggressive optimization strategy. As a stopping criterion, the minimal relative function value change is limited to 10^{-5} and the maximal number of iterations is limited by 200.

In the final evaluation of the optimized design and during the robust topology optimization, the material and geometry uncertainties obtained in the previous section are considered. To reduce the memory demands for storing the covariance matrix, the random fields are discretized using the truncated EOLE expansion. For the principal Young's modulus 100 modes are considered, for the geometric imperfections 10 modes are considered and for the uncertain material orientation 500 modes are considered. Without these approximations, the memory demands of the covariance matrix would be $100 - 300$ terabytes and hence, are far off practical application.

Application of the deterministic optimization leads to the design depicted in figure 5.10a with the numeric performance values presented in table 5.2 after exceeding 200 iterations. At the final design, the function value change is below $3 \cdot 10^{-5}$. The design is based on a direct transition of the outer load to the supports with very few structural elements. Thereby, it significantly differs from the optimized three-point bending beam with isotropic material (see e.g. [68]). The difference is explained by the superior stiffness in axial direction, which makes structural elements oriented in y-direction less attractive and hence, suppresses the generation of a truss-like structure. In consequence, the design does not focus on improving geometric bending stiffness by putting material at the outer edges, but instead exploits the anisotropic material behavior.

Considering the numeric values, the deterministic compliance is 35% lower than the solution of a simple 3-point bending beam ($c = 13.8$Nm) with a rectangular cross section, the same thickness and the width adapted to get the same volume using Bernoulli beam theory. Due to significant uncertainties in the material data, a high coefficient of variation $\mathrm{CoV} = \frac{\sigma_c}{\mu_c} \approx 10\%$ is obtained using a Monte Carlo simulation with 2500 samples. More detailed additional investigations showed that the uncertain principal Young's modulus E_1 is responsible for approximately 66% of the variance, the uncertain material orientation η for approximately 33% of the variance whereas the remaining uncertainties are responsible for less than 1%. Since the stiffness matrix shows very low nonlineari-

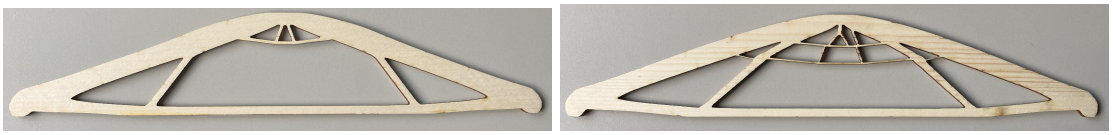


Figure 5.11: Image of the fabricated specimen. Left: deterministically optimized design, Right: robust optimized design.

ties with respect to E_1 and η in the mean case, the first-order second-moment method shows highly accurate results compared to the Monte Carlo simulations. Remaining differences of up to 5% for the standard deviation and 1% for the mean are explained by both, inaccurate Monte Carlo results due to a rather low sample size and remaining nonlinearities in the objective function.

Due to the rather high accuracy of the first-order second-moment method, it is also used as a probabilistic method for the robust design optimization. The modified optimization problem reads

$$\begin{aligned} &\min_{\boldsymbol{\rho}} \mu_c(\boldsymbol{\rho}) + 10\sigma_c(\boldsymbol{\rho}) \\ &\text{s.t.} \\ &0 \leq \rho_i \leq 1 \quad \forall i \\ &\frac{V(\boldsymbol{y}) - 0.3V_0}{0.3V_0} \leq \mathbf{0} \end{aligned} \quad , \tag{5.12}$$

where $\mu_c(\boldsymbol{\rho})$ and $\sigma_c(\boldsymbol{\rho})$ represent the design dependent mean and standard deviation obtained with the first-order second-moment method. The gradients are obtained using the principal sensitivity approach with the relative step size 10^{-4}. All remaining optimization parameters and methods are equal to the deterministic optimization.

The robust design optimization is terminated after 200 iterations where the relative function value change is below $3 \cdot 10^{-5}$. The final design is depicted in figure 5.10b with the numeric values presented in 5.2. In difference to the deterministic design, the robust design contains more details based on smaller features. Using this strategy, the redundant structures are less influenced by the short-correlated component of uncertain Young's moduli. Besides, more material is transferred to the center, where uncertainties of the angle η are lower. In consequence, the standard deviation drops by $2-3\%$ compared to the deterministic design. However, unexpectedly, also the deterministic function value is improved, which means that the robust optimization also leads to an improved local minimum. In consequence, also the mean value obtained with the Monte Carlo simulation drops by 1%. Again, the first-order second-moment approximation shows good accuracy with 5% error for the standard deviation and 1% error for the standard deviation, confirming that it is a suitable approach for the current example.

Finally, the robust design optimization leads to more robust results, although the differences are very low. These findings are to be confirmed by the following experiments.

5.2.3 Experimental assessment of the optimized structure

The optimized designs (figures 5.10a and 5.10b) are fabricated using a laser cutting machine. The cutting paths are generated from the density field by computation of the contour line with 50% density. To obtain a smooth geometry instead of a stepped contour, the density values are assigned

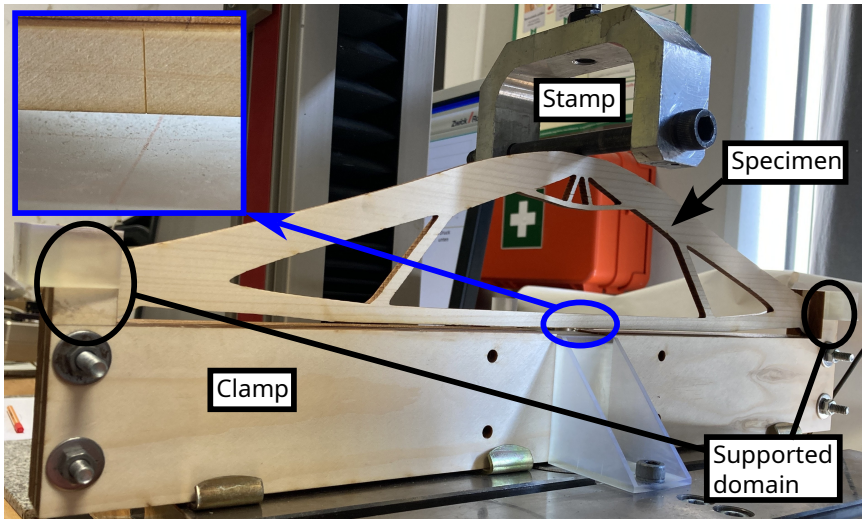


Figure 5.12: Experimental setup for the assessment of the optimized designs. Close-up: The centers of specimen and clamp are marked to reduce positioning errors.

to the element centers, whereas the intermediate values are interpolated from the center values using linear interpolation. Two fabricated models are depicted in figure 5.11. In total, 48 specimens were fabricated from 24 wooden sheets. For a fair comparison, always one deterministically optimized design and one robust optimized design is cut from the same sheet.

The experimental setup is shown in figure 5.12. It consists of the compression machine "Zwick/Roell Z010" which is equipped with the force sensor "Zwick/Roell Xforce HP", a stamp, the wooden clamp and the specimen. Displacements are measured using the internal position sensor of the machine. The clamp is designed based on the problem description shown in figure 5.1 and also fabricated using a laser cutting machine. It consists of stabilizing walls that suppress a displacement in z-direction and base plates (hidden by the walls) that suppress displacements in y-direction. The clamp is fabricated such that there is a small contact force $(1 - 5\text{N})$ between specimen and walls in the initial configuration, which is not expected to have a significant influence on the result. Between the two support domains the specimen does not have contact to the clamp, even in the deformed state. The clamp is mounted on the machine base using a 3D-printed bracket, which accurately adjusts the clamp to the machine and presses it on the base. To ensure contact between the clamp and the plate at defined critical points (near the supports), shims are put below the clamp. The load is introduced by the round surface of a screw with a diameter of 10mm to reduce the risk of stress concentrations. Considering the whole experiment design, the stiffness of the experimental setup is much higher than the stiffness of the specimen, leading to low inaccuracies.

All experiments lead to a load displacement curve as shown in figure 5.13. The curve starts at zero force with a low slope. This is caused by effects, such as missing contact between stamp and specimen in the beginning and settlement of the setup. Afterwards, the slope changes to a fixed value for a long distance, until it decreases again. At approximately 80N force, the slope drastically reduces. As shown in the right figure 5.13, this is caused by out-of-plane buckling of the structure. Although out-of-plane imperfections are considered during the robust optimization and the Monte

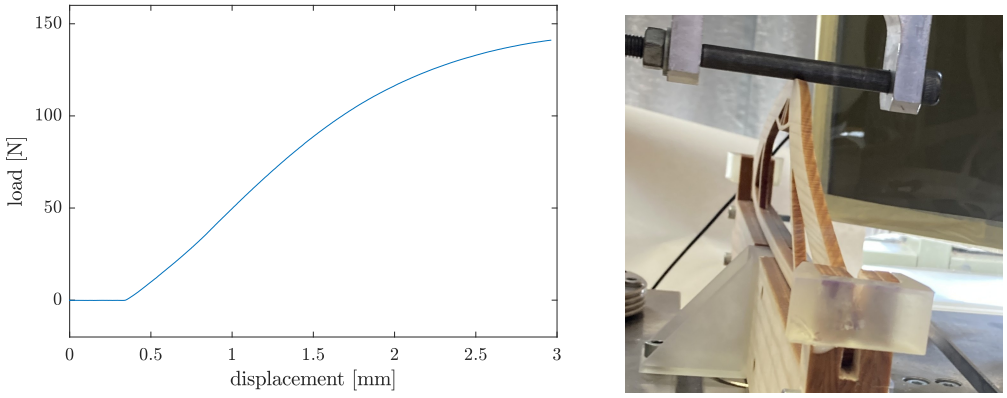
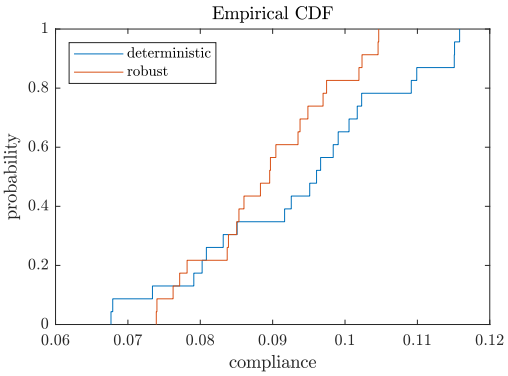


Figure 5.13: Load displacement curve of a single specimen (left) and the deformed shape at the maximal force (right).

Carlo simulations, the buckling is not detected in the simulations, due to the use of linear finite elements. To compare the simulations with the experiments, only the range where the structure behaved linearly (15N to 40N) is considered. The experimental compliance at the considered load of $F = 100$N is afterwards extrapolated from the slope in this range.

Considering the equivalent compliance of all experiments, the cumulative distribution function and the stochastic moments presented in figure 5.14 are obtained. As seen in the cumulative distribution function, the experimental results do not contain any outliers that have to be corrected. Based on the data, the general result from the numeric analysis and optimization is confirmed. The values of the mean values are approximately 90Nmm while the standard deviations are roughly 10Nmm. As obtained in the probabilistic analysis, the design obtained using a robust design optimization performs better regarding both, mean value and standard deviation.

However, the differences between the designs obtained experimentally are much higher than expected from the numeric experiments. Especially for the standard deviation, large differences of 50% between experiment and analysis are causing the robust design to perform much better. There are several possible reasons for this observation. First, the considered sample size of 24 specimens per design is rather low, leading to serious inaccuracies. Second, some inaccuracies in the execution of the experiments may remain, although the experiment is designed to reduce the risk of execution errors. Finally, there might be some effects that are not considered in the simulation. For example, only the principal Young's modulus has been considered to be independently uncertain, whereas also all other material properties might be uncertain too. Similarly, only the imperfect material orientation in the axial-tangential-plane has been estimated from the measured imperfections in the axial-radial plane and some correlations that have not been considered might occur. Finally, also plastic deformations (especially at the load introduction) and imperfections introduced by the manufacturing have not been considered in the simulations, but might have a significant influence on the result. Hence, putting more effort into quantification of occurring uncertainties for the simulations and increasing the sample size is expected to improve the results.



	deterministic optimized	robust optimized
experiments		
μ_c	93.77Nmm	89.20Nmm
σ_c	14.56Nmm	9.56Nmm
Monte Carlo		
μ_c	87.37Nmm	87.01Nmm
σ_c	9.50Nmm	9.35Nmm

Figure 5.14: Left: cumulative distribution function of deterministically and robustly optimized designs. Right: Numeric values of the related stochastic moments.

Chapter 6

Summary and outlook

The current thesis addresses the challenge of scaling robust design optimization from simple, academic examples to more realistic examples with spatially distributed random variables and fine discretizations. Therefore, the focus is set on three crucial aspects:

First, the discretized covariance matrix of a random field becomes extremely memory consuming if a fine discretization is used. Therefore, different existing methods for the discretization of continuous random fields are discussed and the well suited expansion optimal linear estimator method is applied to most examples.

Second, the probabilistic methods are not scalable in many cases. Especially, the computation of the gradients of the robust objective function is challenging and leads to many additional systems of equations to be solved. Therefore, new non-intrusive formulations for the first-order second-moment method and for the second-order fourth-moment method are derived and applied to several examples. The new principal sensitivity first-order second-moment method only requires one additional evaluation of the design response, independent of the number of variables. Due to its non-intrusive characteristics, it does not increase the memory demands compared to a deterministic optimization and may also be applied to closed-source commercial code. In consequence, it is applicable to any example where a deterministic optimization works. However, it is very inaccurate for many problems and hence, the more accurate principal-sensitivity second-order fourth-moment method is presented. As the first-order approach, it does not increase the memory demands. Although it is much less computationally costly than alternative second-order approaches, the number of function evaluations still grows linearly with the number of random variables, limiting its scalability. However, the number of independent random variables may be reduced using the expansion optimal linear estimator method such that the combination of both methods is at least scalable to fine finite element meshes. In some special cases, e.g. buckling of slender structures, also the second-order fourth-moment method shows inaccurate results. Therefore, a new variant of the first-order second-moment method is developed which specifically addresses these kinds of problems. There, the scalability is similar to the second-order fourth-moment method.

Finally, also the finite element simulation might be the bottleneck. To scale the solution of finite element problems to large problems or fine discretizations, the governing systems of equations are solved using iterative methods. The commonly used multigrid method is proven to be scalable, i.e., the required number of vector matrix multiplications is independent of the problem size. However, it does not converge well in the context of highly anisotropic materials, such as composites or wood.

To address this challenge, the problem is analyzed using local Fourier analysis and a new smoothing algorithm for the multigrid method is presented.

Besides the constructed academic examples, all three components are applied to a more realistic example of a wooden three point bending beam. For that, the problem is formulated such that it can actually be fabricated. The considered uncertainties are measured experimentally using compression tests of the material elasticity and image processing for the material orientation. Based on the problem definition, the expansion optimal linear estimator method shows to be a crucial component, since it reduces the memory requirements of the random field from hundreds of terabytes to a few gigabytes at a low approximation error. However, the remaining number of independent random variables stays high, such that only the first-order second-moment method can solve the problem in appropriate time at high accuracy. Application of the robust design optimization leads to an improved result compared to the deterministically optimized design regarding both, stochastic moments and nominal performance. The designs are tested experimentally afterwards, generally confirming the results, but showing up to 50% different results compared to the simulations.

Based on the current work, several options and challenges stay for the future: First, the last example showed that the real occurring uncertainties might differ from the constructed academic examples. Many higher-order methods show good performance in the constructed examples, but may not be applicable at all if the number of independent random variables cannot be reduced to a small value (e.g., lower than 100). Therefore, more focus should be set on realistic problems including uncertainty quantification and experimental validation. Second, as the first-order second-moment method showed to be the only truly scalable method (from the methods considered), more effort should be taken to increase its accuracy. One possible path might be to augment the formulation by analytic expressions, similar to physically informed neural networks. Alternatively, different approaches might be combined. For example, the first-order second-moment method may be combined with the second-order approach, such that only the most important variables are processed by a second-order approach while the remaining variables are processed by a first-order approach. Another possible path is to combine stochastic gradients with the first-order second-moment method. Since the first-order second-moment method usually fails at perfect designs, the imperfect realizations of stochastic gradients might increase the accuracy while preventing the optimizer to only focus on a single imperfection.

Chapter 7

Bibliography

[1] SpaceX, https://www.spacex.com/media/Capabilities&Services.pdf, accessed 2025-06-16 (2025).

[2] M. Shuman, https://www.freightos.com/de/freight-resources/luftfracht-und-luftfrachtversand-luftfrachtgebuehren-preise-kosten-und-angebote/, accessed 2025-06-16 (2025).

[3] S. Grihon, A380 Weight Savings using Numerical Structural Optimisation, in: AAAF Materials for Aerospace Applications 20th AAAF Colloquium, Paris, 2003, pp. 1–10.
URL https://www.researchgate.net/publication/259709151_A380_Weight_Savings_using_Numerical_Structural_Optimisation

[4] J. Ait-Djoudi, https://nachrichten.idw-online.de/2015/09/16/nominierung-fuer-den-deutschen-zukunftspreis-professor-dr-claus-emmelmann-dabei, accessed 2025-06-18 (2015).

[5] C. Emmelmann, D. Herzog, J. Kranz, 10 - Design for laser additive manufacturing, in: M. Brandt (Ed.), Laser Additive Manufacturing, Woodhead Publishing Series in Electronic and Optical Materials, Woodhead Publishing, 2017, pp. 259–279. doi:10.1016/B978-0-08-100433-3.00010-5.
URL https://www.sciencedirect.com/science/article/pii/B9780081004333000105

[6] N. Aage, E. Andreassen, B. S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design, Nature 550 (7674) (2017) 84–86. doi:10.1038/nature23911.
URL https://www.nature.com/articles/nature23911

[7] N. Aage, O. Sigmund, B. B. Lazarov, E. Andreassen, Topwing data (Jun 2020). doi:10.11583/DTU.c.5043437.v1.
URL https://data.dtu.dk/collections/TopWing_Data/5043437/1

[8] I. Elishakoff, Safety Factors and Reliability: Friends or Foes?, Springer, 2004.

[9] P. W. Christensen, A. Klarbring, An introduction to structural optimization, no. 153 in Solid mechanics and its applications, Springer, Dordrecht London, 2009.

[10] L. Harzheim, Strukturoptimierung: Grundlagen und Anwendungen, 2nd Edition, Europa Lehrmittel, Haan-Gruiten, 2014.

[11] M. P. Bendsøe, O. Sigmund, Topology Optimization Theory, Methods, and Applications, 2nd Edition, Springer-Verlag Berlin Heidelberg, 2004.
URL http://www.springer.com/us/book/9783540429920#aboutBook

[12] K. Svanberg, The method of moving asymptotes - a new method for structural optimization, International Journal for Numerical Methods in Engineering 24 (2) (1987) 359–373. doi:10.1002/nme.1620240207.
URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620240207/abstract

[13] K. Svanberg, A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations, SIAM Journal on Optimization 12 (2) (2002) 555–573. doi:10.1137/S1052623499362822.

[14] Fleury Claude, Braibant Vincent, Structural optimization: A new dual method using mixed variables, International Journal for Numerical Methods in Engineering 23 (3) (1986) 409–428. doi:10.1002/nme.1620230307.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620230307

[15] O. Querin, G. Steven, Y. Xie, Evolutionary structural optimisation (ESO) using a bidirectional algorithm, Engineering Computations 15 (8) (1998) 1031–1048. doi:10.1108/02644409810244129.
URL https://doi.org/10.1108/02644409810244129

[16] B. Bourdin, Filters in topology optimization, International Journal for Numerical Methods in Engineering 50 (9) (2001) 2143–2158. doi:10.1002/nme.116.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.116

[17] B. S. Lazarov, O. Sigmund, Filters in topology optimization based on Helmholtz-type differential equations, International Journal for Numerical Methods in Engineering 86 (6) (2011) 765–781. doi:10.1002/nme.3072.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3072

[18] A. Clausen, E. Andreassen, On filter boundary conditions in topology optimization, Structural and Multidisciplinary Optimization 56 (5) (2017) 1147–1155. doi:10.1007/s00158-017-1709-1.
URL https://doi.org/10.1007/s00158-017-1709-1

[19] M. Wallin, N. Ivarsson, O. Amir, D. Tortorelli, Consistent boundary conditions for PDE filter regularization in topology optimization, Structural and Multidisciplinary Optimization (Apr. 2020). doi:10.1007/s00158-020-02556-w.
URL https://doi.org/10.1007/s00158-020-02556-w

[20] J. K. Guest, J. H. Prévost, T. Belytschko, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, International Journal for Numerical Methods in Engineering 61 (2) (2004) 238–254. doi:10.1002/nme.1064.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1064

[21] O. Sigmund, Morphology-based black and white filters for topology optimization, Structural and Multidisciplinary Optimization 33 (4-5) (2007) 401–424. doi:10.1007/s00158-006-0087-x.
URL https://link.springer.com/article/10.1007/s00158-006-0087-x

[22] F. Wang, B. S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Structural and Multidisciplinary Optimization 43 (6) (2011) 767–784. doi:10.1007/s00158-010-0602-y.
URL https://doi.org/10.1007/s00158-010-0602-y

[23] T. Buhl, C. B. W. Pedersen, O. Sigmund, Stiffness design of geometrically nonlinear structures using topology optimization, Structural and Multidisciplinary Optimization 19 (2) (2000) 93–104. doi:10.1007/s001580050089.
URL http://link.springer.com/article/10.1007/s001580050089

[24] L. Scherz, B. Kriegesmann, C. B. W. Pedersen, A condition number-based numerical stabilization method for geometrically nonlinear topology optimization, International Journal for Numerical Methods in Engineering 125 (23) (2024) e7574, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7574. doi:10.1002/nme.7574.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.7574

[25] N. Pedersen, Maximization of eigenvalues using topology optimization, Structural and Multidisciplinary Optimization 20 (1) (2000) 2–11. doi:10.1007/s001580050130.
URL https://doi.org/10.1007/s001580050130

[26] F. Ferrari, B. S. Lazarov, O. Sigmund, Eigenvalue topology optimization via efficient multilevel solution of the frequency response, International Journal for Numerical Methods in Engineering 115 (7) (2018) 872–892, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.5829. doi:10.1002/nme.5829.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5829

[27] P. Duysinx, M. P. Bendsøe, Topology optimization of continuum structures with local stress constraints, International Journal for Numerical Methods in Engineering 43 (1998) 1453–1478.

[28] G. A. da Silva, N. Aage, A. T. Beck, O. Sigmund, Local versus global stress constraint strategies in topology optimization: A comparative study, International Journal for Numerical Methods in Engineering 122 (21) (2021) 6003–6036, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6781. doi:10.1002/nme.6781.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6781

[29] N. Strömberg, A. Klarbring, Topology optimization of structures in unilateral contact, Structural and Multidisciplinary Optimization 41 (1) (2010) 57–64. doi:10.1007/s00158-009-0407-z.
URL https://doi.org/10.1007/s00158-009-0407-z

[30] A. F. de Vito, W. M. Vicente, Y. M. Xie, Topology optimization applied to the core of structural engineered wood product, Structures 48 (2023) 1567–1575. doi:10.1016/j.istruc.2023.01.036.
URL https://www.sciencedirect.com/science/article/pii/S235201242300036X

[31] N. P. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods for structural topology optimization: a review, Structural and Multidisciplinary Optimization 48 (3) (2013) 437–472. doi:10.1007/s00158-013-0912-y.
URL https://doi.org/10.1007/s00158-013-0912-y

[32] F. Wein, P. D. Dunning, J. A. Norato, A review on feature-mapping methods for structural optimization, Structural and Multidisciplinary Optimization 62 (4) (2020) 1597–1638. doi:10.1007/s00158-020-02649-6.
URL https://doi.org/10.1007/s00158-020-02649-6

[33] E. Hardee, K.-H. Chang, J. Tu, K. K. Choi, I. Grindeanu, X. Yu, A CAD-based design parameterization for shape optimization of elastic solids, Advances in Engineering Software 30 (3) (1999) 185–199. doi:10.1016/S0965-9978(98)00065-9.
URL https://www.sciencedirect.com/science/article/pii/S0965997898000659

[34] V. Braibant, C. Fleury, Shape optimal design using B-splines, Computer Methods in Applied Mechanics and Engineering 44 (3) (1984) 247–267. doi:10.1016/0045-7825(84)90132-4.
URL http://www.sciencedirect.com/science/article/pii/0045782584901324

[35] A. D. Belegundu, J. S. Arora, A study of mathematical programming methods for structural optimization. Part I: Theory, International Journal for Numerical Methods in Engineering 21 (9) (1985) 1583–1599, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620210904. doi:10.1002/nme.1620210904.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620210904

[36] R. Reitinger, K.-U. Bletzinger, E. Ramm, Shape optimization of buckling sensitive structures, Computing Systems in Engineering 5 (1) (1994) 65–75. doi:10.1016/0956-0521(94)90038-8.
URL https://www.sciencedirect.com/science/article/pii/0956052194900388

[37] R. T. Haftka, R. V. Grandhi, Structural shape optimization—A survey, Computer Methods in Applied Mechanics and Engineering 57 (1) (1986) 91–106. doi:10.1016/0045-7825(86)90072-1.

[38] O. Pironneau, Optimal Shape Design for Elliptic Systems, Springer, Berlin, Heidelberg, 1984. doi:10.1007/978-3-642-87722-3.
URL http://link.springer.com/10.1007/978-3-642-87722-3

[39] A. D. Belegundu, S. D. Rajan, A shape optimization approach based on natural design variables and shape functions, Computer Methods in Applied Mechanics and Engineering 66 (1) (1988) 87–106. doi:10.1016/0045-7825(88)90061-8.
URL https://www.sciencedirect.com/science/article/pii/0045782588900618

[40] D. A. Tortorelli, A Geometric Representation Scheme Suitable for Shape Optimization, Mechanics of Structures and Machines 21 (1) (1993) 95–121, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/08905459308905182. doi:10.1080/08905459308905182.
URL https://doi.org/10.1080/08905459308905182

[41] M. Firl, R. Wüchner, K.-U. Bletzinger, Regularization of shape optimization problems using FE-based parametrization, Structural and Multidisciplinary Optimization 47 (4) (2013) 507–521. doi:10.1007/s00158-012-0843-z.
URL https://doi.org/10.1007/s00158-012-0843-z

[42] C. Le, T. Bruns, D. Tortorelli, A gradient-based, parameter-free approach to shape optimization, Computer Methods in Applied Mechanics and Engineering 200 (9) (2011) 985–996. doi:10.1016/j.cma.2010.10.004.
URL http://www.sciencedirect.com/science/article/pii/S0045782510002823

[43] E. Stavropoulou, M. Hojjat, K.-U. Bletzinger, In-plane mesh regularization for node-based shape optimization problems, Computer Methods in Applied Mechanics and Engineering 275 (2014) 39–54. doi:10.1016/j.cma.2014.02.013.
URL https://www.sciencedirect.com/science/article/pii/S004578251400070X

[44] I. Antonau, S. Warnakulasuriya, K.-U. Bletzinger, F. M. Bluhm, M. Hojjat, R. Wüchner, Latest developments in node-based shape optimization using Vertex Morphing parameterization, Structural and Multidisciplinary Optimization 65 (7) (2022) 198. doi:10.1007/s00158-022-03279-w.
URL https://doi.org/10.1007/s00158-022-03279-w

[45] I. Antonau, M. Hojjat, K.-U. Bletzinger, Relaxed gradient projection algorithm for constrained node-based shape optimization, Structural and Multidisciplinary Optimization 63 (4) (2021) 1633–1651. doi:10.1007/s00158-020-02821-y.
URL https://doi.org/10.1007/s00158-020-02821-y

[46] A. N. Christiansen, J. A. Bærentzen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, Combined shape and topology optimization of 3D structures, Computers & Graphics 46 (2015) 25–35. doi:10.1016/j.cag.2014.09.021.
URL https://www.sciencedirect.com/science/article/pii/S0097849314001095

[47] D. Schmölz, B. Devresse, A. Geiser, K.-U. Bletzinger, Simultaneous node-based shape and thickness optimization of thin-walled structures using the explicit Vertex Morphing method, Structural and Multidisciplinary Optimization 68 (2) (2025). doi:10.1007/s00158-025-03977-1.

[48] J. Stegmann, E. Lund, Discrete material optimization of general composite shell structures, International Journal for Numerical Methods in Engineering 62 (14) (2005) 2009–2027, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1259. doi:10.1002/nme.1259.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1259

[49] S. N. Sørensen, R. Sørensen, E. Lund, DMTO – a method for Discrete Material and Thickness Optimization of laminated composite structures, Structural and Multidisciplinary Optimization 50 (1) (2014) 25–47. doi:10.1007/s00158-014-1047-5.
URL https://doi.org/10.1007/s00158-014-1047-5

[50] M.-P. Schmidt, L. Couret, C. Gout, C. B. W. Pedersen, Structural topology optimization with smoothly varying fiber orientations, Structural and Multidisciplinary Optimization 62 (6) (2020) 3105–3126. doi:10.1007/s00158-020-02657-6.
URL https://doi.org/10.1007/s00158-020-02657-6

[51] M. Montemurro, A. Mas, S.-e. Zerrouq, Topology and anisotropy optimisation of continua using non-uniform rational basis spline entities, Computer Methods in Applied Mechanics and Engineering 420 (2024) 116714. doi:10.1016/j.cma.2023.116714.
URL https://www.sciencedirect.com/science/article/pii/S004578252300837X

[52] M. Baandrup, O. Sigmund, H. Polk, N. Aage, Closing the gap towards super-long suspension bridges using computational morphogenesis, Nature Communications 11 (1) (2020) 2735, number: 1 Publisher: Nature Publishing Group. doi:10.1038/s41467-020-16599-6.
URL https://www.nature.com/articles/s41467-020-16599-6

[53] T. Borrvall, J. Petersson, Large-scale topology optimization in 3D using parallel computing, Computer Methods in Applied Mechanics and Engineering 190 (46) (2001) 6201–6229. doi:10.1016/S0045-7825(01)00216-X.
URL https://www.sciencedirect.com/science/article/pii/S004578250100216X

[54] S. Wang, E. d. Sturler, G. H. Paulino, Large-scale topology optimization using preconditioned Krylov subspace methods with recycling, International Journal for Numerical Methods in Engineering 69 (12) (2007) 2441–2468, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1798. doi:10.1002/nme.1798.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1798

[55] O. Amir, M. P. Bendsøe, O. Sigmund, Approximate reanalysis in topology optimization, International Journal for Numerical Methods in Engineering 78 (12) (2009) 1474–1491, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2536. doi:10.1002/nme.2536.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2536

[56] O. Amir, O. Sigmund, On reducing computational effort in topology optimization: how far can we go?, Structural and Multidisciplinary Optimization 44 (1) (2011) 25–29. doi:10.1007/s00158-010-0586-7.
URL http://link.springer.com/10.1007/s00158-010-0586-7

[57] U. Trottenberg, C. W. Oosterlee, A. Schüller, A. Brandt, P. Oswald, K. Stüben, Multigrid, transferred to digital print Edition, Elsevier Academic Press, Amsterdam Heidelberg, 2007.

[58] S. R. Fulton, P. E. Ciesielski, W. H. Schubert, Multigrid Methods for Elliptic Problems: A Review, Monthly Weather Review 114 (5) (1986) 943–959, publisher: American Meteorological Society Section: Monthly Weather Review. doi:10.1175/1520-0493(1986)114<0943:MMFEPA>2.0.CO;2.
URL https://journals.ametsoc.org/view/journals/mwre/114/5/1520-0493_1986_114_0943_mmfepa_2_0_co_2.xml

[59] T. Dreyer, B. Maar, V. Schulz, Multigrid optimization in applications, Journal of Computational and Applied Mathematics 120 (1) (2000) 67–84. doi:10.1016/S0377-0427(00)00304-6.
URL https://www.sciencedirect.com/science/article/pii/S0377042700003046

[60] O. Amir, N. Aage, B. S. Lazarov, On multigrid-CG for efficient topology optimization, Structural and Multidisciplinary Optimization 49 (5) (2014) 815–829. doi:10.1007/s00158-013-1015-5.
URL https://doi.org/10.1007/s00158-013-1015-5

[61] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, Structural and Multidisciplinary Optimization 43 (1) (2011) 1–16. doi:10.1007/s00158-010-0594-7.
URL https://link.springer.com/article/10.1007/s00158-010-0594-7

[62] N. Aage, E. Andreassen, B. S. Lazarov, Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework, Structural and Multidisciplinary Optimization 51 (3) (2015) 565–572. doi:10.1007/s00158-014-1157-0.
URL https://doi.org/10.1007/s00158-014-1157-0

[63] E. A. Träff, O. Sigmund, N. Aage, Topology optimization of ultra high resolution shell structures, Thin-Walled Structures 160 (2021) 107349. doi:10.1016/j.tws.2020.107349.
URL http://www.sciencedirect.com/science/article/pii/S026382312031212X

[64] O. Amir, Efficient stress-constrained topology optimization using inexact design sensitivities, International Journal for Numerical Methods in Engineering 122 (13) (2021) 3241–3272, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6662. doi:10.1002/nme.6662.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6662

[65] O. Amir, One-shot procedures for efficient minimum compliance topology optimization, Structural and Multidisciplinary Optimization 67 (3) (2024) 39. doi:10.1007/s00158-024-03763-5.
URL https://doi.org/10.1007/s00158-024-03763-5

[66] D. Peetz, A. Elbanna, On the use of multigrid preconditioners for topology optimization, Structural and Multidisciplinary Optimization 63 (2) (2021) 835–853. doi:10.1007/s00158-020-02750-w.
URL https://doi.org/10.1007/s00158-020-02750-w

[67] N. Aage, B. S. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, Structural and Multidisciplinary Optimization 47 (4) (2013) 493–505. doi:10.1007/s00158-012-0869-2.
URL https://doi.org/10.1007/s00158-012-0869-2

[68] F. Ferrari, O. Sigmund, A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D, Structural and Multidisciplinary Optimization 62 (4) (2020) 2211–2228. doi:10.1007/s00158-020-02629-w.
URL https://doi.org/10.1007/s00158-020-02629-w

[69] I. Elishakoff, An Idea of the Uncertainty Triangle, The Shock and Vibration Digest 22 (1990) 1.

[70] A. D. Kiureghian, O. Ditlevsen, Aleatory or epistemic? Does it matter?, Structural Safety 31 (2) (2009) 105–112. doi:10.1016/j.strusafe.2008.06.020.
URL https://www.sciencedirect.com/science/article/pii/S0167473008000556

[71] I. Papaioannou, M. Daub, M. Drieschner, F. Duddeck, M. Ehre, L. Eichner, M. Eigel, M. Götz, W. Graf, L. Grasedyck, R. Gruhlke, D. Hömberg, M. Kaliske, D. Moser, Y. Petryna, D. Straub, Assessment and design of an engineering structure with

polymorphic uncertainty quantification, GAMM-Mitteilungen 42 (2) (2019) e201900009, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/gamm.201900009. doi:10.1002/gamm.201900009.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201900009

[72] H. Hederberg, C.-J. Thore, Worst-case compliance for independently constrained uncertain loads, Computers & Structures 289 (2023) 107178. doi:10.1016/j.compstruc.2023.107178.
URL https://www.sciencedirect.com/science/article/pii/S0045794923002080

[73] M. Kegl, B. Brank, B. Harl, M. M. Oblak, Efficient handling of stability problems in shell optimization by asymmetric 'worst-case' shape imperfection, International Journal for Numerical Methods in Engineering 73 (9) (2008) 1197–1216, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2113. doi:https://doi.org/10.1002/nme.2113.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2113

[74] C. Dapogny, F. Iutzeler, A. Meda, B. Thibert, Entropy-regularized Wasserstein distributionally robust shape and topology optimization, Structural and Multidisciplinary Optimization 66 (3) (2023) 42. doi:10.1007/s00158-023-03500-4.
URL https://doi.org/10.1007/s00158-023-03500-4

[75] Z. Luo, L.-P. Chen, J. Yang, Y.-Q. Zhang, K. Abdel-Malek, Fuzzy tolerance multilevel approach for structural topology optimization, Computers and Structures 84 (3-4) (2006) 127–140. doi:10.1016/j.compstruc.2005.10.001.

[76] H. Yin, D. Yu, B. Xia, Reliability-based topology optimization for structures using fuzzy set model, Computer Methods in Applied Mechanics and Engineering 333 (2018) 197–217. doi:10.1016/j.cma.2018.01.019.
URL https://www.sciencedirect.com/science/article/pii/S0045782518300215

[77] M. Fina, C. Bisagni, Buckling design optimization of tow-steered composite panels and cylindrical shells considering aleatory and epistemic uncertainties, Computational Mechanics (Feb. 2025). doi:10.1007/s00466-024-02589-8.
URL https://doi.org/10.1007/s00466-024-02589-8

[78] J. Zheng, S. Ding, C. Jiang, Z. Wang, Concurrent topology optimization for thermoelastic structures with random and interval hybrid uncertainties, International Journal for Numerical Methods in Engineering 123 (4) (2022) 1078–1097, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6889. doi:10.1002/nme.6889.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6889

[79] B. Kriegesmann, On the Applicability of First-Order Approximations for Design Optimization under Uncertainty, in: N. Challamel, J. Kaplunov, I. Takewaki (Eds.), Modern Trends in Structural and Solid Mechanics 3 - Non-deterministic mechanics, John Wiley & Sons, Ltd, 2021, pp. 39–60, section: 3 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119831839.ch3. doi:10.1002/9781119831839.ch3.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119831839.ch3

[80] G.-J. Park, T.-H. Lee, K. H. Lee, K.-H. Hwang, Robust Design: An Overview, AIAA Journal 44 (1) (2006) 181–191. doi:10.2514/1.13639.
URL https://doi.org/10.2514/1.13639

[81] G. I. Schuëller, H. A. Jensen, Computational methods in optimization considering uncertainties – An overview, Computer Methods in Applied Mechanics and Engineering 198 (1) (2008) 2–13. doi:10.1016/j.cma.2008.05.004.
URL http://www.sciencedirect.com/science/article/pii/S0045782508002028

[82] Y. Kanno, On three concepts in robust design optimization: absolute robustness, relative robustness, and less variance, Structural and Multidisciplinary Optimization 62 (2) (2020) 979–1000. doi:10.1007/s00158-020-02503-9.
URL https://doi.org/10.1007/s00158-020-02503-9

[83] W. Hu, S. Cheng, J. Yan, J. Cheng, X. Peng, H. Cho, I. Lee, Reliability-based design optimization: a state-of-the-art review of its methodologies, applications, and challenges, Structural and Multidisciplinary Optimization 67 (9) (2024) 168. doi:10.1007/s00158-024-03884-x.
URL https://doi.org/10.1007/s00158-024-03884-x

[84] H.-G. Beyer, B. Sendhoff, Robust optimization – A comprehensive survey, Computer Methods in Applied Mechanics and Engineering 196 (33) (2007) 3190–3218. doi:10.1016/j.cma.2007.03.003.
URL http://www.sciencedirect.com/science/article/pii/S0045782507001259

[85] G. Taguchi, Introduction to quality engineering: designing quality into products and processes, Quality Resources, 1986.
URL https://trid.trb.org/view/1179550

[86] G. A. da Silva, E. L. Cardoso, A. T. Beck, Comparison of robust, reliability-based and non-probabilistic topology optimization under uncertain loads and stress constraints, Probabilistic Engineering Mechanics 59 (2020) 103039. doi:10.1016/j.probengmech.2020.103039.
URL http://www.sciencedirect.com/science/article/pii/S0266892020300242

[87] N. Metropolis, S. Ulam, The Monte Carlo Method, Journal of the American Statistical Association 44 (247) (1949) 335. doi:10.2307/2280232.
URL https://www.jstor.org/stable/2280232?origin=crossref

[88] M. Schevenels, B. S. Lazarov, O. Sigmund, Robust topology optimization accounting for spatially varying manufacturing errors, Computer Methods in Applied Mechanics and Engineering 200 (49–52) (2011) 3613–3627. doi:10.1016/j.cma.2011.08.006.
URL http://www.sciencedirect.com/science/article/pii/S0045782511002611

[89] M. Jansen, G. Lombaert, M. Diehl, B. S. Lazarov, O. Sigmund, M. Schevenels, Robust topology optimization accounting for misplacement of material, Structural and Multidisciplinary Optimization 47 (3) (2013) 317–333. doi:10.1007/s00158-012-0835-z.
URL https://doi.org/10.1007/s00158-012-0835-z

[90] K. Luo, X. He, H. Jing, Topology optimization of bridges under random traffic loading using stochastic reduced-order models, Probabilistic Engineering Mechanics 75 (2024) 103583. doi:10.1016/j.probengmech.2024.103583.
URL https://www.sciencedirect.com/science/article/pii/S0266892024000055

[91] Z. Meng, Z. Zhang, H. Zhou, H. Chen, B. Yu, Robust design optimization of imperfect stiffened shells using an active learning method and a hybrid surrogate model, Engineering Optimization 52 (12) (2020) 2044–2061, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0305215X.2019.1702978. doi:10.1080/0305215X.2019.1702978.
URL https://doi.org/10.1080/0305215X.2019.1702978

[92] V. Dubourg, B. Sudret, J.-M. Bourinet, Reliability-based design optimization using kriging surrogates and subset simulation, Structural and Multidisciplinary Optimization 44 (5) (2011) 673–690. doi:10.1007/s00158-011-0653-8.
URL https://link.springer.com/article/10.1007/s00158-011-0653-8

[93] M. Papadrakakis, V. Papadopoulos, N. D. Lagaros, Structural reliability analyis of elastic-plastic structures using neural networks and Monte Carlo simulation, Computer Methods in Applied Mechanics and Engineering 136 (1) (1996) 145–163. doi:10.1016/0045-7825(96)01011-0.
URL https://www.sciencedirect.com/science/article/pii/0045782596010110

[94] T. Chatterjee, S. Chakraborty, R. Chowdhury, A Critical Review of Surrogate Assisted Robust Design Optimization, Archives of Computational Methods in Engineering 26 (1) (2019) 245–274. doi:10.1007/s11831-017-9240-5.
URL http://link.springer.com/10.1007/s11831-017-9240-5

[95] V. Keshavarzzadeh, F. Fernandez, D. A. Tortorelli, Topology optimization under uncertainty via non-intrusive polynomial chaos expansion, Computer Methods in Applied Mechanics and Engineering 318 (2017) 120–147. doi:10.1016/j.cma.2017.01.019.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782516313019

[96] M. Tootkaboni, A. Asadpoure, J. K. Guest, Topology optimization of continuum structures under uncertainty – A Polynomial Chaos approach, Computer Methods in Applied Mechanics and Engineering 201-204 (2012) 263–275. doi:10.1016/j.cma.2011.09.009.
URL https://www.sciencedirect.com/science/article/pii/S0045782511002982

[97] D. Lee, S. Rahman, Robust design optimization under dependent random variables by a generalized polynomial chaos expansion, Structural and Multidisciplinary Optimization 63 (5) (2021) 2425–2457. doi:10.1007/s00158-020-02820-z.
URL https://doi.org/10.1007/s00158-020-02820-z

[98] Y. Shi, Q. Lan, X. Lan, J. Wu, T. Yang, B. Wang, Robust optimization design of a flying wing using adjoint and uncertainty-based aerodynamic optimization approach, Structural and Multidisciplinary Optimization 66 (5) (2023) 110. doi:10.1007/s00158-023-03559-z.
URL https://doi.org/10.1007/s00158-023-03559-z

[99] S. De, K. Maute, A. Doostan, Topology optimization under microscale uncertainty using stochastic gradients, Structural and Multidisciplinary Optimization 66 (1) (2022) 17. doi:10.1007/s00158-022-03417-4.
URL https://doi.org/10.1007/s00158-022-03417-4

[100] S. De, J. Hampton, K. Maute, A. Doostan, Topology optimization under uncertainty using a stochastic gradient-based approach, Structural and Multidisciplinary Optimization 62 (5)

(2020) 2255–2278. doi:10.1007/s00158-020-02599-z.
URL https://doi.org/10.1007/s00158-020-02599-z

[101] M. Grieshammer, L. Pflug, M. Stingl, A. Uihlein, The continuous stochastic gradient method: part I–convergence theory, Computational Optimization and Applications 87 (3) (2024) 935–976. doi:10.1007/s10589-023-00542-8.
URL https://doi.org/10.1007/s10589-023-00542-8

[102] A. Mousavi, A. Uihlein, L. Pflug, E. Wadbro, Topology optimization of broadband acoustic transition section: a comparison between deterministic and stochastic approaches, Structural and Multidisciplinary Optimization 67 (5) (2024) 67. doi:10.1007/s00158-024-03784-0.
URL https://doi.org/10.1007/s00158-024-03784-0

[103] A. Uihlein, The Continuous Stochastic Gradient Method, Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2024).
URL https://open.fau.de/handle/openfau/30533

[104] S. Shen, D. Zhang, X. Han, C. Jiang, Q. Li, A robust optimization framework for design of robotic system with kinematic and dynamic criteria, Computer Methods in Applied Mechanics and Engineering 423 (2024) 116866. doi:10.1016/j.cma.2024.116866.
URL https://www.sciencedirect.com/science/article/pii/S0045782524001221

[105] B. S. Lazarov, M. Schevenels, O. Sigmund, Robust design of large-displacement compliant mechanisms, Mechanical Sciences 2 (2) (2011) 175–182, publisher: Copernicus GmbH. doi:10.5194/ms-2-175-2011.
URL https://ms.copernicus.org/articles/2/175/2011/

[106] B. S. Lazarov, M. Schevenels, O. Sigmund, Topology optimization considering material and geometric uncertainties using stochastic collocation methods, Structural and Multidisciplinary Optimization 46 (4) (2012) 597–612. doi:10.1007/s00158-012-0791-7.
URL http://link.springer.com/article/10.1007/s00158-012-0791-7

[107] J. Zhao, C. Wang, Robust topology optimization under loading uncertainty based on linear elastic theory and orthogonal diagonalization of symmetric matrices, Computer Methods in Applied Mechanics and Engineering 273 (2014) 204–218. doi:10.1016/j.cma.2014.01.018.
URL http://www.sciencedirect.com/science/article/pii/S0045782514000310

[108] I. Doltsinis, Z. Kang, Robust design of structures using optimization methods, Computer Methods in Applied Mechanics and Engineering 193 (23) (2004) 2221–2237. doi:10.1016/j.cma.2003.12.055.
URL http://www.sciencedirect.com/science/article/pii/S0045782504000787

[109] I. Doltsinis, Z. Kang, G. Cheng, Robust design of non-linear structures using optimization methods, Computer Methods in Applied Mechanics and Engineering 194 (12) (2005) 1779–1795. doi:10.1016/j.cma.2004.02.027.
URL http://www.sciencedirect.com/science/article/pii/S0045782504004165

[110] B. S. Lazarov, M. Schevenels, O. Sigmund, Topology optimization with geometric uncertainties by perturbation techniques, International Journal for Numerical Methods in Engineering 90 (11) (2012) 1321–1336. doi:10.1002/nme.3361.
URL http://onlinelibrary.wiley.com/doi/10.1002/nme.3361/abstract

[111] G. A. da Silva, E. L. Cardoso, Stress-based topology optimization of continuum structures under uncertainties, Computer Methods in Applied Mechanics and Engineering 313 (2017) 647–672. doi:10.1016/j.cma.2016.09.049.
URL http://www.sciencedirect.com/science/article/pii/S0045782516312816

[112] K.-I. Ypsilantis, M. G. R. Faes, N. D. Lagaros, N. Aage, D. Moens, Robust topology and discrete fiber orientation optimization under principal material uncertainty, Computers & Structures 300 (2024) 107421. doi:10.1016/j.compstruc.2024.107421.
URL https://www.sciencedirect.com/science/article/pii/S0045794924001500

[113] K.-H. Lee, G.-J. Park, Robust optimization considering tolerances of design variables, Computers & Structures 79 (1) (2001) 77–86. doi:10.1016/S0045-7949(00)00117-6.
URL http://www.sciencedirect.com/science/article/pii/S0045794900001176

[114] B. Kriegesmann, J. K. Lüdeker, Robust compliance topology optimization using the first-order second-moment method, Structural and Multidisciplinary Optimization 60 (1) (2019) 269–286. doi:10.1007/s00158-019-02216-8.
URL https://doi.org/10.1007/s00158-019-02216-8

[115] M. Kranz, J. K. Lüdeker, B. Kriegesmann, A generalized approach for robust topology optimization using the first-order second-moment method for arbitrary response functions, Structural and Multidisciplinary Optimization 66 (5) (2023) 98. doi:10.1007/s00158-023-03540-w.
URL https://doi.org/10.1007/s00158-023-03540-w

[116] T. Schmidt, B. Kriegesmann, R. Seifried, Robust contact-constrained topology optimization considering uncertainty at the contact support, Structural and Multidisciplinary Optimization 67 (4) (2024) 46. doi:10.1007/s00158-024-03750-w.
URL https://doi.org/10.1007/s00158-024-03750-w

[117] K. Steltner, B. Kriegesmann, C. B. W. Pedersen, Robust sizing optimization of stiffened panels subject to geometric imperfections using fully nonlinear postbuckling analyses, Thin-Walled Structures 175 (2022) 109195. doi:10.1016/j.tws.2022.109195.
URL https://www.sciencedirect.com/science/article/pii/S0263823122001719

[118] G. Bletsos, L. Radtke, T. Rung, Adjoint-assisted robust shape optimization of an idealized arterial bypass graft using the FOSM method, Structural and Multidisciplinary Optimization 68 (6) (2025) 120. doi:10.1007/s00158-025-04050-7.
URL https://doi.org/10.1007/s00158-025-04050-7

[119] K. B. Fragkos, E. M. Papoutsis-Kiachagias, K. C. Giannakoglou, pFOSM: An efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products, Computers & Fluids 181 (2019) 57–66. doi:10.1016/j.compfluid.2019.01.016.
URL http://www.sciencedirect.com/science/article/pii/S0045793018307680

[120] L. Komini, M. Langelaar, B. Kriegesmann, Robust topology optimization considering part distortion and process variability in additive manufacturing, Advances in Engineering Software 186 (2023) 103551. doi:10.1016/j.advengsoft.2023.103551.
URL https://www.sciencedirect.com/science/article/pii/S0965997823001424

[121] K. Steltner, C. B. W. Pedersen, B. Kriegesmann, Semi-intrusive approach for stiffness and strength topology optimization under uncertainty, Optimization and Engineering (Sep. 2022). doi:10.1007/s11081-022-09770-z.
URL https://doi.org/10.1007/s11081-022-09770-z

[122] M. Jansen, G. Lombaert, M. Schevenels, Robust topology optimization of structures with imperfect geometry based on geometric nonlinear analysis, Computer Methods in Applied Mechanics and Engineering 285 (2015) 452–467. doi:10.1016/j.cma.2014.11.028.
URL http://www.sciencedirect.com/science/article/pii/S004578251400454X

[123] T. Nishino, J. Kato, Robust topology optimization based on finite strain considering uncertain loading conditions, International Journal for Numerical Methods in Engineering 122 (6) (2021) 1427–1455. doi:10.1002/nme.6584.
URL https://onlinelibrary.wiley.com/doi/10.1002/nme.6584

[124] R. T. Haftka, Z. Gürdal, Elements of Structural Optimization, Vol. 11 of Solid Mechanics And Its Applications, Springer Netherlands, Dordrecht, 1992. doi:10.1007/978-94-011-2550-5.
URL http://link.springer.com/10.1007/978-94-011-2550-5

[125] P. Michaleris, D. A. Tortorelli, C. A. Vidal, Tangent operators and design sensitivity formulations for transient non-linear coupled problems with applications to elastoplasticity, International Journal for Numerical Methods in Engineering 37 (14) (1994) 2471–2499. doi:10.1002/nme.1620371408.
URL http://onlinelibrary.wiley.com/doi/10.1002/nme.1620371408/abstract

[126] M. P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Archive of Applied Mechanics 69 (9-10) (1999) 635–654. doi:10.1007/s004190050248.
URL http://link.springer.com/10.1007/s004190050248

[127] M. Stolpe, K. Svanberg, An alternative interpolation scheme for minimum compliance topology optimization, Structural and Multidisciplinary Optimization 22 (2) (2001) 116–124. doi:10.1007/s001580100129.
URL https://doi.org/10.1007/s001580100129

[128] T. E. Bruns, O. Sigmund, D. A. Tortorelli, Numerical methods for the topology optimization of structures that exhibit snap-through, International Journal for Numerical Methods in Engineering 55 (10) (2002) 1215–1237, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.544. doi:https://doi.org/10.1002/nme.544.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.544

[129] C. Forbes, M. Evans, N. Hastings, B. Peacock, Statistical distributions, 4th Edition, John Wiley & Sons, 2011.
URL https://onlinelibrary.wiley.com/doi/10.1002/9780470627242

[130] A. Haldar, S. Mahadevan, Probability, Reliability and Statistical Methods in Engineering Design, 1st Edition, John Wiley & Sons, New York ; Chichester England, 1999.

[131] W. R. Pestman, Mathematical Statistics, De Gruyter, 2009. doi:10.1515/9783110208535.
URL https://www.degruyterbrill.com/document/doi/10.1515/9783110208535/html

[132] R. J. Adler, J. E. Taylor, Random Fields and Geometry, Springer Monographs in Mathematics, Springer, New York, NY, 2007, iSSN: 1439-7382. doi:10.1007/978-0-387-48116-6.
URL http://link.springer.com/10.1007/978-0-387-48116-6

[133] B. Sudret, A. Der Kiureghian, Stochastic finite element methods and reliability: a state-of-the-art report, Tech. Rep. UCB/SEMM-2000/08, Department of Civil and Environmental Engineering, University of California Berkeley, Berkeley, California (2000).

[134] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005. doi:10.7551/mitpress/3206.001.0001.
URL https://direct.mit.edu/books/book/2320/Gaussian-Processes-for-Machine-Learning

[135] N. Cressie, Statistics for Spatial Data, John Wiley & Sons, 1993, google-Books-ID: MzN_BwAAQBAJ. doi:âŤť10.1002/9781119115151.

[136] A. Der Kiureghian, J.-B. Ke, The stochastic finite element method in structural reliability, Probabilistic Engineering Mechanics 3 (2) (1988) 83–91. doi:10.1016/0266-8920(88)90019-7.
URL https://www.sciencedirect.com/science/article/pii/0266892088900197

[137] H. G. Matthies, C. E. Brenner, C. G. Bucher, C. Guedes Soares, Uncertainties in probabilistic numerical analysis of structures and solids-Stochastic finite elements, Structural Safety 19 (3) (1997) 283–336. doi:10.1016/S0167-4730(97)00013-1.
URL https://www.sciencedirect.com/science/article/pii/S0167473097000131

[138] C.-C. Li, A. Der Kiureghian, Optimal Discretization of Random Fields, Journal of Engineering Mechanics 119 (6) (1993) 1136–1154, publisher: American Society of Civil Engineers. doi:10.1061/(ASCE)0733-9399(1993)119:6(1136).
URL https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9399%281993%29119%3A6%281136%29

[139] T. Lehmann, W. Oberschelp, E. Pelikan, R. Repges, Die Karhunen-Loève-Transformation, in: T. Lehmann, W. Oberschelp, E. Pelikan, R. Repges (Eds.), Bildverarbeitung für die Medizin: Grundlagen, Modelle, Methoden, Anwendungen, Springer, Berlin, Heidelberg, 1997, pp. 295–307. doi:10.1007/978-3-642-60487-4_12.
URL https://doi.org/10.1007/978-3-642-60487-4_12

[140] O. Ditlevsen, Random Field Interpolation Between Point by Point Measured Properties, in: P. D. Spanos, C. A. Brebbia (Eds.), Computational Stochastic Mechanics, Springer Netherlands, Dordrecht, 1991, pp. 801–812. doi:10.1007/978-94-011-3692-1_67.
URL https://doi.org/10.1007/978-94-011-3692-1_67

[141] M. Baitsch, D. Hartmann, Optimization of slender structures considering geometrical imperfections, Inverse Problems in Science and Engineering 14 (6) (2006) 623–637. doi:10.1080/17415970600573494.
URL https://doi.org/10.1080/17415970600573494

[142] K. Kolanek, S. Jendo, Random field models of geometrically imperfect structures with "clamped" boundary conditions, Probabilistic Engineering Mechanics 23 (2-3) (2008) 219–226. doi:10.1016/j.probengmech.2007.12.002.
URL https://linkinghub.elsevier.com/retrieve/pii/S0266892007000549

[143] C. L. Lemieux, Monte Carlo and Quasi-Monte Carlo Sampling, Springer Series in Statistics, Springer, New York, NY, 2009. doi:10.1007/978-0-387-78165-5.
URL https://link.springer.com/10.1007/978-0-387-78165-5

[144] B. Kriegesmann, R. Rolfes, C. Hühne, A. Kling, Fast Probabilistic Design Procedure for Axially Compressed Composite Cylinders, Composites Structures 93 (2011) 3140–3149. doi:10.1016/j.compstruct.2011.06.017.

[145] J. C. Krüger, M. Kranz, T. Schmidt, R. Seifried, B. Kriegesmann, An efficient and non-intrusive approach for robust design optimization with the first-order second-moment method, Computer Methods in Applied Mechanics and Engineering 414 (2023) 116136. doi:10.1016/j.cma.2023.116136.
URL https://linkinghub.elsevier.com/retrieve/pii/S0045782523002608

[146] O. Ditlevsen, Comments on "first-order second-moment approximation in reliability of structural systems: critical review and alternative approach", by Krzysztof Dolinski, Structural Safety 3 (1) (1985) 59–61. doi:10.1016/0167-4730(85)90008-6.
URL http://www.sciencedirect.com/science/article/pii/0167473085900086

[147] K. Dolinski, First-order second-moment approximation in reliability of structural systems: Critical review and alternative approach, Structural Safety 1 (3) (1982) 211–231. doi:10.1016/0167-4730(82)90027-3.
URL http://www.sciencedirect.com/science/article/pii/0167473082900273

[148] J. C. Krüger, B. Kriegesmann, Robust design optimization using a non-intrusive second-order approximation of stochastic moments, Structural and Multidisciplinary Optimization 67 (7) (2024) 124. doi:10.1007/s00158-024-03843-6.
URL https://doi.org/10.1007/s00158-024-03843-6

[149] B. Kriegesmann, J. C. Krüger, K. Steltner, T. Schmidt, R. Seifried, Robust design optimization of structures with multimodal responses using Taylor expansions, Structural and Multidisciplinary Optimization - submitted (2025).

[150] M. Rosenblatt, Remarks on a Multivariate Transformation, The Annals of Mathematical Statistics 23 (3) (1952) 470–472, publisher: Institute of Mathematical Statistics. doi:10.1214/aoms/1177729394.

[151] P. Wriggers, Nonlinear Finite Element Method, Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-71001-1_1.
URL http://link.springer.com/chapter/10.1007/978-3-540-71001-1_1

[152] J. C. Krüger, B. Kriegesmann, On the application of the multigrid method to topology optimization with orthotropic material with varying orientation, Structural and Multidisciplinary Optimization 68 (8) (2025) 158. doi:10.1007/s00158-025-04102-y.
URL https://doi.org/10.1007/s00158-025-04102-y

[153] D. W. Green, J. E. Winandy, D. E. Kretschmann, Mechanical properties of wood, Wood handbook : wood as an engineering material. Madison, WI : USDA Forest Service, Forest Products Laboratory, 1999. General technical report FPL ; GTR-113: Pages 4.1-4.45 113 (1999).
URL https://research.fs.usda.gov/treesearch/7149

[154] J. C. Krüger, Uncertainty quantification of spruce wood: Compresson tests and images of material orientation (2026). doi:10.15480/882.15956.

[155] C. Jenkel, F. Leichsenring, W. Graf, M. Kaliske, Stochastic modelling of uncertainty in timber engineering, Engineering Structures 99 (2015) 296–310. doi:10.1016/j.engstruct.2015.04.049.
URL https://www.sciencedirect.com/science/article/pii/S0141029615003053

[156] A. Price, I. A mathematical discussion on the structure of wood in relation to its elastic properties, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 228 (659-669) (1929) 1–62. doi:10.1098/rsta.1929.0001.
URL https://royalsocietypublishing.org/doi/10.1098/rsta.1929.0001

[157] F. J. Barthold, E. Stein, A continuum mechanical-based formulation of the variational sensitivity analysis in structural optimization. Part I: analysis, Structural optimization 11 (1) (1996) 29–42. doi:10.1007/BF01279652.
URL https://doi.org/10.1007/BF01279652

[158] M. Kleiber, T. D. Hien, H. Antunez, P. Kowalczyk, Parameter sensitivity in nonlinear mechanics: theory and finite element computations, John Wiley, Chichester [England]; New York, 1997, oCLC: 45728284.
URL http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=17910

[159] J. K. Lüdeker, O. Sigmund, B. Kriegesmann, Inverse homogenization using isogeometric shape optimization, Computer Methods in Applied Mechanics and Engineering 368 (2020) 113170. doi:10.1016/j.cma.2020.113170.
URL http://www.sciencedirect.com/science/article/pii/S0045782520303558

Appendix A

Gradients for nonlinear finite elements

Using nonlinear finite elements, the computation of many response functions and their gradients become more complicated. Especially the shape gradient is much more challenging than gradients with respect to the topology in simple linear topology optimization problems. Since these gradients are used in several examples and the literature (e.g. [157] [158]) regarding this topic is rather abstract, a detailed derivation for the gradient of the inner force is found in the following. A overview regarding the continuum mechanical background can be found in [151].

A.1 Basics from continuum mechanics

When looking at nonlinear finite elements, the equilibrium is considered for the deformed state. Hence kinematics that describe the relations between deformed configuration and reference configuration are required. In the following, lowercase letters correspond to the deformed configuration and uppercase letters to the reference configuration.

A.1.1 Kinematics and equilibrium

The position of a mass point is given by the vector $\boldsymbol{X}$ (reference) or $\boldsymbol{x}$ (deformed). There is a bijective function $\varphi(\boldsymbol{X})$ such that $\boldsymbol{x} = \varphi(\boldsymbol{X}) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X})$ with the deformation vector $\boldsymbol{u}(\boldsymbol{X})$. The gradient of φ is called the deformation gradient $\boldsymbol{F}$ and is computed by

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \quad . \tag{A.1}$$

A similar deformation measure is the displacement gradient $\boldsymbol{H}$ given by

$$\boldsymbol{H} = \frac{d\boldsymbol{u}}{d\boldsymbol{X}} = \boldsymbol{F} - \boldsymbol{1} \quad . \tag{A.2}$$

Using these definitions, the weak form of the equilibrium is given by

$$\int \mathrm{Div}(\boldsymbol{P})^T \boldsymbol{\eta} dV + \int \rho_0 \boldsymbol{f}_b^T \boldsymbol{\eta} dV = 0 \tag{A.3}$$

where $\boldsymbol{\eta}$ is a test function, $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor and dV represents a volume element in the reference configuration. The body forces $\boldsymbol{f}_b$ and the density ρ_0 represent the external loads. Using integration by parts, the whole term simplifies to

$$\int \boldsymbol{S} : \delta\boldsymbol{E} dV - \int \rho_0 \boldsymbol{f}_b^T \boldsymbol{\eta} dV - \int \boldsymbol{t}^T \boldsymbol{\eta} dA = 0 \tag{A.4}$$

with the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the general strain $\boldsymbol{E}$. The resulting outer surface loads $\boldsymbol{t} = \boldsymbol{P}^T \boldsymbol{N}$ are computed from the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ and the surface normal $\boldsymbol{N}$.

A.1.2 Constitutive law and strain measure

Both, constitutive laws and strain measures are not defined uniquely. In the current thesis, the St. Venant-Krichhoff material law given by

$$\boldsymbol{S} = \boldsymbol{\mathcal{C}} : \boldsymbol{E} \tag{A.5}$$

using the material tensor $\boldsymbol{\mathcal{C}}$ is considered. Using Nye notation on stresses and strains, the material law is reformulated to the more handy vector-matrix form

$$\boldsymbol{S}_e = \mathbb{C} \cdot \boldsymbol{E}_e \tag{A.6}$$

with the material matrix $\mathbb{C}$ (similar to Hooke's law) and the strain in Nye notation $\boldsymbol{E}_e$. Note that the St. Venant-Kirchhoff constitutive is only valid for small strains.

Independent of the constitutive law, different strain measures are available. In the current thesis, the Green-Lagrange strain tensor $\boldsymbol{E}$ given by

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{1}) = \frac{1}{2}(\boldsymbol{H} + \boldsymbol{H}^T + \boldsymbol{H}^T\boldsymbol{H}) \tag{A.7}$$

is used. It is able to handle rigid body motion, but fails fore large elastic formulations (i.e. large strains). This limitation is not critical since the consecutive law is already limited to small strains.

A.1.3 Finite element notation and inner forces

In the finite element method, the spatial properties are computed from nodal values using ansatz functions. There are different potential ways to write down the nodal displacements and positions since these are 3D properties. Hence different notations are defined in the following.

- Nodal displacements of element e written as matrix $\boldsymbol{u}_e = \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ \vdots & \vdots & \vdots \end{pmatrix}$
- Nodal displacements written as vector $\hat{\boldsymbol{u}}_e = \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{2,1} & u_{2,2} & u_{2,3} & \dots \end{pmatrix}^T$
- Nodal coordinates of element e written as matrix $\boldsymbol{X}_e = \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} \\ X_{2,1} & X_{2,2} & X_{2,3} \\ \vdots & \vdots & \vdots \end{pmatrix}$

- Nodal coordinates written as vector $\hat{\boldsymbol{X}}_e = \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} & X_{2,1} & X_{2,2} & X_{2,3} & \dots \end{pmatrix}^T$

The ansatz functions $N_I(\boldsymbol{X})$ are built such that $p(\boldsymbol{X}) = \sum_I N_I(\boldsymbol{X}) \cdot p_I$ with the property p. These ansatz functions are collected in a matrix such that

$$p(\boldsymbol{X}) = \boldsymbol{N}\boldsymbol{p} = \begin{pmatrix} N_1(\boldsymbol{X}) & N_2(\boldsymbol{X}) & \dots \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \end{pmatrix} \quad . \tag{A.8}$$

This collection also works, if the property is vector valued. In this case the vectors must be written row-wise like the nodal displacement matrix $\boldsymbol{u}_e$ leading to

$$\boldsymbol{u}(\boldsymbol{X}) = (\boldsymbol{N}\boldsymbol{u}_e)^T \quad . \tag{A.9}$$

If the vector valued property is written like $\hat{\boldsymbol{u}}_e$, than a modified matrix of ansatz functions defined as

$$\hat{\boldsymbol{N}}(\boldsymbol{X}) = \begin{pmatrix} N_1(\boldsymbol{X}) & 0 & 0 & N_2(\boldsymbol{X}) & 0 & 0 & \dots \\ 0 & N_1(\boldsymbol{X}) & 0 & 0 & N_2(\boldsymbol{X}) & 0 & \dots \\ 0 & 0 & N_1(\boldsymbol{X}) & 0 & 0 & N_2(\boldsymbol{X}) & \dots \end{pmatrix} \tag{A.10}$$

leading to $\boldsymbol{u}(\boldsymbol{X}) = \hat{\boldsymbol{N}}(\boldsymbol{X})\hat{\boldsymbol{u}}_e$ is used. The integration is done for each element separately and assembled afterwards. Using the ansatz function $\hat{\boldsymbol{N}}$ the test function $\boldsymbol{\eta}$ is defined as

$$\boldsymbol{\eta}(\boldsymbol{X}) = \hat{\boldsymbol{N}}(\boldsymbol{X})\delta\hat{\boldsymbol{u}}_e \quad . \tag{A.11}$$

Since the node displacements are constant over the volume, they can be taken out of the integral (A.4) leading to

$$\boldsymbol{R}(\boldsymbol{u}) - \boldsymbol{P} = \boldsymbol{0} \tag{A.12}$$

with the inner forces

$$\boldsymbol{R} = \bigcup_e \boldsymbol{R}_e = \bigcup_e \int_{V_e} \frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e}^T \boldsymbol{S}_e dV \tag{A.13}$$

and the outer forces

$$\boldsymbol{P} = \bigcup_e \boldsymbol{P}_e = \bigcup_e \int_{V_e} \hat{\boldsymbol{N}} \rho_0 \boldsymbol{f}_b dV + \bigcup_e \int_{V_e} \hat{\boldsymbol{N}} \boldsymbol{t} dA \quad , \tag{A.14}$$

where $\bigcup_e$ represents the assembly over all elements and $\boldsymbol{E}_e$ and $\boldsymbol{S}_e$ represent vector (e.g. Nye) notations of the strains and stresses. The integrals are usually solved using Gauss-Legendre integration on isoparametric elements. Hence the inner force reads

$$\boldsymbol{R}_e = \sum_i \Omega_i \left. \frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e}^T \right|_{\boldsymbol{r}_i} \boldsymbol{S}_e(\boldsymbol{r}_i) \cdot \det(\boldsymbol{J}_e) \tag{A.15}$$

with the Jacobi matrix $\boldsymbol{J}_e$

$$\boldsymbol{J}_e(\tilde{\boldsymbol{X}}) = \frac{\partial \boldsymbol{X}}{\partial \tilde{\boldsymbol{X}}} = \frac{\partial \boldsymbol{N}}{\partial \tilde{\boldsymbol{X}}} \cdot \boldsymbol{X}_e \quad , \tag{A.16}$$

the Gaussian weights Ω_i and the Gauss Points $\boldsymbol{r}_i$.

A.2 Inner loads

Using the previous defined notation, kinematics and constitutive laws, the inner load $\boldsymbol{R}_e$ obtained by a finite element formulation (A.15) reads

$$\boldsymbol{R}_e = \sum_i \Omega_i \left.\frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e}\right|_{\boldsymbol{r}_i} \boldsymbol{\mathcal{C}}\boldsymbol{E}_e(\boldsymbol{r}_i) \cdot \det(\boldsymbol{J}_e) \quad . \tag{A.17}$$

The Green-Lagrange strain $\boldsymbol{E}_e$ is computed from the strain tensor

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{1}) = \frac{1}{2}\left(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\boldsymbol{x}_e\boldsymbol{x}_e^T\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T - \boldsymbol{1}\right) \tag{A.18}$$

with $\hat{\boldsymbol{x}_e} = \hat{\boldsymbol{X}}_e + \hat{\boldsymbol{u}_e}$ and the deformation gradient $\boldsymbol{F}$ computed by

$$\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\boldsymbol{x}_e = \boldsymbol{F}^T = \begin{pmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_2}{\partial X_1} & \frac{\partial x_3}{\partial X_1} \\ \frac{\partial x_1}{\partial X_2} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_3}{\partial X_2} \\ \frac{\partial x_1}{\partial X_3} & \frac{\partial x_2}{\partial X_3} & \frac{\partial x_3}{\partial X_3} \end{pmatrix} \quad . \tag{A.19}$$

The strain vector in Nye notation is afterwards found as

$$\boldsymbol{E}_e = \begin{pmatrix} E_{11} & E_{22} & E_{33} & 2E_{12} & 2E_{13} & 2E_{23} \end{pmatrix}^T \quad . \tag{A.20}$$

The gradient of the Green-Lagrange strain is computed by differentiation of (A.18) with respect to deformation matrix entry (i, j) with

$$\frac{\partial \boldsymbol{E}}{\partial u_{i,j}} = \frac{1}{2}\left(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{x}_e}{\partial u_{i,j}}\boldsymbol{F} + \boldsymbol{F}^T\frac{\partial \boldsymbol{x}_e}{\partial u_{i,j}}^T\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T\right) \quad , \tag{A.21}$$

which can be further simplified to

$$\frac{\partial \boldsymbol{E}}{\partial u_{i,j}} = \frac{1}{2}\left(\frac{\partial N_i}{\partial \boldsymbol{X}}\boldsymbol{F}_{j,:} + \boldsymbol{F}_{j,:}^T\frac{\partial N_i}{\partial \boldsymbol{X}}^T\right) \quad , \tag{A.22}$$

where $\boldsymbol{F}_{j,:}$ represents the j-th row of $\boldsymbol{F}$. The transpose operator is applied after extracting the row. In the 3D case, this reads

$$\frac{\partial \boldsymbol{E}}{\partial u_{i,j}} = \frac{1}{2}\begin{pmatrix} \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,3} \\ \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,3} \\ \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,3} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,3} \\ \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,3} \\ \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,1} & \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,2} & \frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,3} \end{pmatrix}^T \quad . \tag{A.23}$$

The gradient of the strain in Nye notation is found by taking specific entries

$$\frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e} = \begin{pmatrix} \frac{\partial \mathrm{diag}(\boldsymbol{E})}{\partial \hat{\boldsymbol{u}}_e} \\ \hline \frac{\partial \begin{pmatrix} 2E_{12} & 2E_{13} & 2E_{23} \end{pmatrix}^T}{\partial \hat{\boldsymbol{u}}_e} \end{pmatrix} = \boldsymbol{B} \quad . \tag{A.24}$$

The whole term also relates to the nonlinear version of the "B-matrix" known from linear finite elements. Using the Hadamard product $\odot$, these entries are efficiently computed by

$$\frac{\partial \mathrm{diag}(\boldsymbol{E})}{\partial u_{ei,j}} = \begin{pmatrix} \frac{1}{2}\left(\frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,1} + \boldsymbol{F}_{j,1}^T \frac{\partial N_i}{\partial X_1}^T\right) \\ \frac{1}{2}\left(\frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,2} + \boldsymbol{F}_{j,2}^T \frac{\partial N_i}{\partial X_2}^T\right) \\ \frac{1}{2}\left(\frac{\partial N_i}{\partial X_3}\boldsymbol{F}_{j,3} + \boldsymbol{F}_{j,3}^T \frac{\partial N_i}{\partial X_3}^T\right) \end{pmatrix} = \frac{\partial N_i}{\partial \boldsymbol{X}} \odot \boldsymbol{F}_{j,:}^T \tag{A.25}$$

and

$$\begin{aligned} \frac{\partial \begin{pmatrix} 2E_{12} & 2E_{13} & 2E_{23} \end{pmatrix}^T}{\partial u_{ei,j}} &= \begin{pmatrix} \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,2} + \boldsymbol{F}_{j,1}^T \frac{\partial N_i}{\partial X_2}^T \\ \frac{\partial N_i}{\partial X_1}\boldsymbol{F}_{j,3} + \boldsymbol{F}_{j,1}^T \frac{\partial N_i}{\partial X_3}^T \\ \frac{\partial N_i}{\partial X_2}\boldsymbol{F}_{j,3} + \boldsymbol{F}_{j,2}^T \frac{\partial N_i}{\partial X_3}^T \end{pmatrix} \\ &= \frac{\partial N_i}{\partial \begin{pmatrix} X_1 & X_1 & X_2 \end{pmatrix}^T} \odot \boldsymbol{F}_{j,(2,3,3)}^T + \frac{\partial N_i}{\partial \begin{pmatrix} X_2 & X_3 & X_3 \end{pmatrix}^T} \odot \boldsymbol{F}_{j,(1,1,2)}^T \end{aligned} . \tag{A.26}$$

A.2.1 Gradient of inner load w.r.t. displacement - tangential stiffness matrix

The gradient of the inner forces is often called the tangential stiffness matrix and is obtained by direct differentiation

$$\boldsymbol{K}_e = \sum_i \det(\boldsymbol{J}_e)\Omega_i \left[\left.\frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{u}}_e}\right|_{\boldsymbol{r}_i} \boldsymbol{\mathcal{C}}\boldsymbol{E}_e(\boldsymbol{r}_i) + \left.\frac{\partial \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e}\right|_{\boldsymbol{r}_i} \boldsymbol{\mathcal{C}} \left.\frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e}\right|_{\boldsymbol{r}_i} \right] \quad , \tag{A.27}$$

where $\boldsymbol{E}_e$ is computed by (A.20) and $\left.\boldsymbol{\mathcal{C}}\frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{u}}_e}\right|_{\boldsymbol{r}_i}$ is obtained from (A.24). The remaining term is simplified by $\frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{u}}_e}\boldsymbol{\mathcal{C}}\boldsymbol{E}_e = \frac{\partial^2 \boldsymbol{E}^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{u}}_e} : \boldsymbol{S}$, such that the second-order gradient of the strain is required only for the matrix notation. It is computed by

$$\frac{\partial^2 \boldsymbol{E}}{\partial u_{ei,j} \partial u_{ek,l}} = \frac{1}{2}\left(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{x}_e}{\partial u_{ei,j}} \frac{\partial \boldsymbol{x}_e}{\partial u_{ek,l}}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T + \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{x}_e}{\partial u_{ek,l}} \frac{\partial \boldsymbol{x}_e}{\partial u_{ei,j}}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T \right) \tag{A.28}$$

which can be simplified to

$$\frac{\partial^2 \boldsymbol{E}}{\partial u_{ei,j} \partial u_{ek,l}} = \begin{cases} \frac{\partial N_i}{\partial \boldsymbol{X}} \frac{\partial N_k}{\partial \boldsymbol{X}}^T & j = l \\ 0 & j \neq l \end{cases} . \tag{A.29}$$

Computing the dyadic product in 3D leads to

$$\frac{\partial^2 \boldsymbol{E}}{\partial u_{ei,j} \partial u_{ek,j}} = \frac{\partial N_i}{\partial \boldsymbol{X}} \frac{\partial N_k}{\partial \boldsymbol{X}}^T = \begin{pmatrix} \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_1} & \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_2} & \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_3} \\ \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_1} & \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_2} & \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_3} \\ \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_1} & \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_2} & \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_3} \end{pmatrix} \tag{A.30}$$

The whole term $\frac{\partial^2 \boldsymbol{E}}{\partial u_{ei,j} \partial u_{ek,j}} : \boldsymbol{S}$ results in

$$\begin{aligned}\frac{\partial^2 \boldsymbol{E}}{\partial u_{ei,j} \partial u_{ek,j}} : \boldsymbol{S} &= \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_1} S_{11} + \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_2} S_{12} + \frac{\partial N_i}{\partial X_1}\frac{\partial N_k}{\partial X_3} S_{13} \\ &+ \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_1} S_{21} + \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_2} S_{22} + \frac{\partial N_i}{\partial X_2}\frac{\partial N_k}{\partial X_3} S_{23} \\ &+ \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_1} S_{31} + \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_2} S_{32} + \frac{\partial N_i}{\partial X_3}\frac{\partial N_k}{\partial X_3} S_{33} \\ &= \frac{dN_i}{d\boldsymbol{X}}^T \begin{pmatrix} \frac{\partial N_k}{\partial X_1} S_{11} + \frac{\partial N_k}{\partial X_2} S_{12} + \frac{\partial N_k}{\partial X_3} S_{13} \\ \frac{\partial N_k}{\partial X_1} S_{21} + \frac{\partial N_k}{\partial X_2} S_{22} + \frac{\partial N_k}{\partial X_3} S_{23} \\ \frac{\partial N_k}{\partial X_1} S_{31} + \frac{\partial N_k}{\partial X_2} S_{32} + \frac{\partial N_k}{\partial X_3} S_{33} \end{pmatrix} = \frac{dN_i}{d\boldsymbol{X}}^T \boldsymbol{S} \frac{dN_k}{d\boldsymbol{X}}\end{aligned} \quad . \tag{A.31}$$

Defining

$$\frac{\partial^2 \boldsymbol{E}}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{u}}_e} : \boldsymbol{S} = \boldsymbol{K}_Q \tag{A.32}$$

in addition with some index processing finally leads to the simple formulation

$$\begin{aligned}\boldsymbol{K}_Q(1{:}3{:}\text{end},1{:}3{:}\text{end}) &= \frac{d\boldsymbol{N}}{d\boldsymbol{X}}^T \boldsymbol{S} \frac{d\boldsymbol{N}}{d\boldsymbol{X}} \\ \boldsymbol{K}_Q(2{:}3{:}\text{end},2{:}3{:}\text{end}) &= \frac{d\boldsymbol{N}}{d\boldsymbol{X}}^T \boldsymbol{S} \frac{d\boldsymbol{N}}{d\boldsymbol{X}} \\ \boldsymbol{K}_Q(3{:}3{:}\text{end},3{:}3{:}\text{end}) &= \frac{d\boldsymbol{N}}{d\boldsymbol{X}}^T \boldsymbol{S} \frac{d\boldsymbol{N}}{d\boldsymbol{X}}\end{aligned} \quad , \tag{A.33}$$

where $\boldsymbol{K}_Q(1{:}3{:}\text{end},1{:}3{:}\text{end})$ means that every third row and every third column is considered.

A.2.2 gradient of inner loads with respect to nodal positions

Differentiation of the inner forces leads (A.15) to

$$\frac{\partial \boldsymbol{R}_e}{\partial \hat{\boldsymbol{X}}_e} = \sum_i \det(\boldsymbol{J}_e)\Omega_i \frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} \boldsymbol{S}_e + \sum_i \det(\boldsymbol{J}_e)\Omega_i \frac{\partial \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e} \boldsymbol{\mathcal{C}} \frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{X}}_e} + \sum_i \Omega_i \frac{\partial \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e} \boldsymbol{S}_e \frac{\partial \det(\boldsymbol{J}_e)}{\partial \hat{\boldsymbol{X}}_e}^T \tag{A.34}$$

where $\boldsymbol{S}_e$ is the stress in Nye notation. To shorten the notation, several terms in (A.35) are substituted such that

$$\frac{\partial \boldsymbol{R}_e}{\partial \hat{\boldsymbol{X}}_e} = \sum_i \det(\boldsymbol{J}_e)\Omega_i T_1 + \sum_i \det(\boldsymbol{J}_e)\Omega_i \frac{\partial \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e} \boldsymbol{\mathcal{C}} T_2 + \sum_i \Omega_i T_3 \tag{A.35}$$

with

$$T_1 = \frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} \boldsymbol{S}_e \tag{A.36}$$

$$T_2 = \frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{X}}_e} \tag{A.37}$$

$$T_3 = \frac{\partial \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e} \boldsymbol{S}_e \frac{\partial \det(\boldsymbol{J}_e)}{\partial \hat{\boldsymbol{X}}_e}^T \quad . \tag{A.38}$$

T_1 is rewritten as

$$T_1 = \frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} \boldsymbol{S}_e = \frac{\partial^2 \boldsymbol{E}}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} : \boldsymbol{S} \quad . \tag{A.39}$$

The derivative of the Green Lagrange strain is given by

$$\begin{aligned} 2\frac{\partial^2 \boldsymbol{E}}{\partial u_{i,j} \partial X_{k,l}} &= \frac{\partial^2 (\boldsymbol{H}^T + \boldsymbol{H} + \boldsymbol{H}^T \boldsymbol{H})}{\partial u_{i,j} \partial X_{k,l}} \\ &= \frac{\partial^2 \boldsymbol{H}^T}{\partial u_{i,j} \partial X_{k,l}} + \frac{\partial^2 \boldsymbol{H}}{\partial u_{i,j} \partial X_{k,l}} + \frac{\partial^2 \boldsymbol{H}^T}{\partial u_{i,j} \partial X_{k,l}} \boldsymbol{H} + \boldsymbol{H}^T \frac{\partial^2}{\partial u_{i,j} \partial X_{k,l}} \\ &+ \frac{\partial \boldsymbol{H}^T}{\partial u_{i,j}} \frac{\partial \boldsymbol{H}}{\partial X_{k,l}} + \frac{\partial \boldsymbol{H}^T}{\partial X_{k,l}} \frac{\partial \boldsymbol{H}}{\partial u_{i,j}} \\ &= \frac{\partial^2 \boldsymbol{H}^T}{\partial u_{i,j} \partial X_{k,l}} (\boldsymbol{1} + \boldsymbol{H}) + (\boldsymbol{1} + \boldsymbol{H}^T) \frac{\partial^2 \boldsymbol{H}}{\partial u_{i,j} \partial X_{k,l}} \\ &+ \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{u}_e}{\partial u_{i,j}} \boldsymbol{u}_e^T \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X} \partial X_{k,l}} + \frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{N} \partial X_{k,l}} \boldsymbol{u}_e \frac{\partial \boldsymbol{u}_e^T}{\partial u_{i,j}} \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X}} \end{aligned} \tag{A.40}$$

leading to the stress strain product

$$\begin{aligned} 2\frac{\partial^2 \boldsymbol{E}}{\partial u_{i,j} \partial X_{k,l}} : \boldsymbol{S} &= (\boldsymbol{F}^T \frac{\partial^2 \boldsymbol{H}}{\partial u_{i,j} \partial X_{k,l}}) : \boldsymbol{S} + (\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{u}_e}{\partial u_{i,j}} \boldsymbol{u}_e^T \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X} \partial X_{ek,l}}) : \boldsymbol{S} \\ &= \frac{\partial^2 \boldsymbol{H}}{\partial u_{i,j} \partial X_{k,l}} : (\boldsymbol{F}\boldsymbol{S}) + (\frac{\partial N_i}{\partial \boldsymbol{X}} \boldsymbol{u}_{:,j}^T \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X} \partial X_{k,l}}) : \boldsymbol{S} \end{aligned} \tag{A.41}$$

where $\boldsymbol{u}_{e:,j}$ is the j-th column of $\boldsymbol{u}_e$. The first part is computed by

$$\frac{\partial^2 \boldsymbol{H}}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} : (\boldsymbol{F}\boldsymbol{S}) == - \begin{pmatrix} \boldsymbol{F}\boldsymbol{S} \frac{\partial N_1}{\partial \boldsymbol{X}} \frac{\partial N_1}{\partial \boldsymbol{X}}^T & \boldsymbol{F}\boldsymbol{S} \frac{\partial N_2}{\partial \boldsymbol{X}} \frac{\partial N_1}{\partial \boldsymbol{X}}^T & \dots \\ \boldsymbol{F}\boldsymbol{S} \frac{\partial N_1}{\partial \boldsymbol{X}} \frac{\partial N_2}{\partial \boldsymbol{X}}^T & \boldsymbol{F}\boldsymbol{S} \frac{\partial N_2}{\partial \boldsymbol{X}} \frac{\partial N_2}{\partial \boldsymbol{X}}^T & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad . \tag{A.42}$$

The second part is computed using $\frac{\partial^2 \boldsymbol{N}}{\partial X_{ek,l} \partial \boldsymbol{X}} = -\frac{\partial N_k}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{N}}{\partial X_l}$ from [159] which results in

$$(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{u}_e}{\partial u_{i,j}} \boldsymbol{u}_e^T \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X} \partial X_{k,l}}) : \boldsymbol{S} = -(\frac{\partial N_i}{\partial \boldsymbol{X}} \boldsymbol{u}_{:,j}^T \frac{\partial \boldsymbol{N}^T}{\partial X_l} \frac{\partial N_k}{\partial \boldsymbol{X}}^T) : \boldsymbol{S} = -H_{l,j} \frac{\partial N_k}{\partial \boldsymbol{X}}^T \boldsymbol{S} \frac{\partial N_i}{\partial \boldsymbol{X}} \quad . \tag{A.43}$$

Collecting the node wise quantities leads to

$$(\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{u}_e}{\partial u_{ei,:}} \boldsymbol{u}_e^T \frac{\partial^2 \boldsymbol{N}^T}{\partial \boldsymbol{X} \partial X_{ek,:}}) : \boldsymbol{S} = -(\frac{\partial N_k}{\partial \boldsymbol{X}}^T \boldsymbol{S} \frac{\partial N_i}{\partial \boldsymbol{X}}) \boldsymbol{H} \quad . \tag{A.44}$$

Hence the full term T_1 reads

$$T_1 = \frac{\partial^2 \boldsymbol{E}_e^T}{\partial \hat{\boldsymbol{u}}_e \partial \hat{\boldsymbol{X}}_e} \boldsymbol{S}_e = - \begin{pmatrix} \boldsymbol{F}\boldsymbol{S} \frac{\partial N_1}{\partial \boldsymbol{X}} \frac{\partial N_1}{\partial \boldsymbol{X}}^T & \boldsymbol{F}\boldsymbol{S} \frac{\partial N_2}{\partial \boldsymbol{X}} \frac{\partial N_1}{\partial \boldsymbol{X}}^T & \dots \\ \boldsymbol{F}\boldsymbol{S} \frac{\partial N_1}{\partial \boldsymbol{X}} \frac{\partial N_2}{\partial \boldsymbol{X}}^T & \boldsymbol{F}\boldsymbol{S} \frac{\partial N_2}{\partial \boldsymbol{X}} \frac{\partial N_2}{\partial \boldsymbol{X}}^T & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} - (\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T \boldsymbol{S} \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}) \otimes \boldsymbol{H} \tag{A.45}$$

The part T_2 is computed from differentiating $\boldsymbol{E}$ by

$$\begin{aligned} 2\frac{\partial \boldsymbol{E}}{\partial X_{k,l}} = &\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{X}\partial X_{k,l}}\boldsymbol{x}_e\boldsymbol{x}_e^T\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T + \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\boldsymbol{x}_e\boldsymbol{x}_e^T\frac{\partial^2 \boldsymbol{N}}{\partial \boldsymbol{X}\partial X_{k,l}}^T \\ &+ \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{x}_e}{\partial X_{k,l}}\boldsymbol{x}_e^T\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T + \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\boldsymbol{x}_e\frac{\partial \boldsymbol{x}_e^T}{\partial X_{k,l}}\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}^T \end{aligned} \quad . \tag{A.46}$$

Using $\frac{\partial^2 \boldsymbol{N}}{\partial X_{k,l}\partial \boldsymbol{X}} = -\frac{\partial N_k}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{N}}{\partial X_l}$ leads to

$$\begin{aligned} 2\frac{\partial \boldsymbol{E}}{\partial X_{k,l}} &= -\frac{\partial N_k}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{N}}{\partial X_l}\boldsymbol{x}_e\boldsymbol{F} - \boldsymbol{F}^T\boldsymbol{x}_e^T\frac{\partial \boldsymbol{N}}{\partial X_l}^T\frac{\partial N_k}{\partial \boldsymbol{X}}^T + \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{x}_e}{\partial X_{k,l}}\boldsymbol{F} + \boldsymbol{F}^T\frac{\partial \boldsymbol{x}_e^T}{\partial X_{k,l}}\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} \\ &= -\frac{\partial N_k}{\partial \boldsymbol{X}}\left(\frac{\partial \boldsymbol{N}}{\partial X_l}\boldsymbol{x}_e - \boldsymbol{e}_l\right)\boldsymbol{F} - \boldsymbol{F}^T\left(\frac{\partial \boldsymbol{N}}{\partial X_l}\boldsymbol{x}_e - \boldsymbol{e}_l\right)^T\frac{\partial N_k}{\partial \boldsymbol{X}}^T \\ &= -\frac{\partial N_k}{\partial \boldsymbol{X}}\boldsymbol{H}_l^T\boldsymbol{F} - \boldsymbol{F}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial \boldsymbol{X}}^T \end{aligned} \tag{A.47}$$

where $\boldsymbol{e}_l$ is the l-th unit vector and $\boldsymbol{H}_l$ corresponds to the l-th column of the displacement gradient. Hence the Nye notation reads

$$2\frac{\partial \boldsymbol{E}_e}{\partial X_{ek,l}} = -\begin{pmatrix} \frac{\partial N_k}{\partial X_1}\boldsymbol{H}_l^T\boldsymbol{F}_{:,1} + \boldsymbol{F}_{:,1}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_1}^T \\ \frac{\partial N_k}{\partial X_2}\boldsymbol{H}_l^T\boldsymbol{F}_{:,2} + \boldsymbol{F}_{:,2}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_2}^T \\ \frac{\partial N_k}{\partial X_3}\boldsymbol{H}_l^T\boldsymbol{F}_{:,3} + \boldsymbol{F}_{:,3}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_3}^T \\ \frac{\partial N_k}{\partial X_1}\boldsymbol{H}_l^T\boldsymbol{F}_{:,2} + \boldsymbol{F}_{:,1}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_2}^T \\ \frac{\partial N_k}{\partial X_1}\boldsymbol{H}_l^T\boldsymbol{F}_{:,3} + \boldsymbol{F}_{:,1}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_3}^T \\ \frac{\partial N_k}{\partial X_2}\boldsymbol{H}_l^T\boldsymbol{F}_{:,3} + \boldsymbol{F}_{:,2}^T\boldsymbol{H}_l\frac{\partial N_k}{\partial X_3}^T \end{pmatrix} = -\boldsymbol{B}_k\boldsymbol{H}_l \tag{A.48}$$

where $\boldsymbol{B}_k$ represents the columns of the B-matrix belonging to the k-th node. In 3D it would be the columns $3(k-1)+1$ to $3k$. The term T_2 is now computed from

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{X}_{ek}} = -\boldsymbol{B}_k\boldsymbol{u}_e^T\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{X}} = -\boldsymbol{B}_k\boldsymbol{H} \tag{A.49}$$

leading to

$$\frac{\partial \boldsymbol{E}_e}{\partial \hat{\boldsymbol{X}}_e} = -\begin{pmatrix}\frac{\partial \boldsymbol{\epsilon}_e}{\partial \boldsymbol{X}_{e1}} & \frac{\partial \boldsymbol{\epsilon}_e}{\partial \boldsymbol{X}_{e2}} & \dots\end{pmatrix} = -\begin{pmatrix}\boldsymbol{B}_1\boldsymbol{H} & \boldsymbol{B}_2\boldsymbol{H} & \dots\end{pmatrix} = -\boldsymbol{B}(\boldsymbol{1}\otimes\boldsymbol{H}) \quad . \tag{A.50}$$

The part T_3 is computed in a similar way as in the case of linear finite elements. The derivative of the Jacobi-determinant is computed using

$$\frac{\partial \boldsymbol{E}}{\partial \hat{\boldsymbol{u}}_e} = \frac{1}{2}\left(\frac{\partial \boldsymbol{H}}{\partial \hat{\boldsymbol{u}}_e} + \frac{\partial \boldsymbol{H}^T}{\partial \hat{\boldsymbol{u}}_e} + \boldsymbol{H}^T\frac{\partial \boldsymbol{H}}{\partial \hat{\boldsymbol{u}}_e} + \frac{\partial \boldsymbol{H}^T}{\partial \hat{\boldsymbol{u}}_e}\boldsymbol{H}\right) = \frac{1}{2}\left(\boldsymbol{F}^T\frac{\partial \boldsymbol{H}}{\partial \hat{\boldsymbol{u}}_e} + \frac{\partial \boldsymbol{H}^T}{\partial \hat{\boldsymbol{u}}_e}\boldsymbol{F}\right) \quad , \tag{A.51}$$

such that the whole term reads

$$T_3 = \frac{\partial \boldsymbol{E}}{\partial \hat{\boldsymbol{u}}_e} : \boldsymbol{S}\frac{\partial \det(\boldsymbol{J}_e)}{\partial \hat{\boldsymbol{X}}_e}^T = \begin{pmatrix}\boldsymbol{F}\boldsymbol{S}\frac{\partial N_1}{\boldsymbol{X}} \\ \boldsymbol{F}\boldsymbol{S}\frac{\partial N_2}{\boldsymbol{X}} \\ \vdots\end{pmatrix}\cdot\begin{pmatrix}\frac{\partial N_1}{\partial \boldsymbol{X}}^T & \frac{\partial N_2}{\partial \boldsymbol{X}}^T & \dots\end{pmatrix}\det(\boldsymbol{J}_e) \quad . \tag{A.52}$$