Non-Linear Shape Optimization Using Local Subspace Projections

Przemyslaw Musialski[1*] Christian Hafner[1*] Florian Rist[1] Michael Birsak[1] Michael Wimmer[1†] Leif Kobbelt[2†]

[1]TU Wien [2]RWTH Aachen

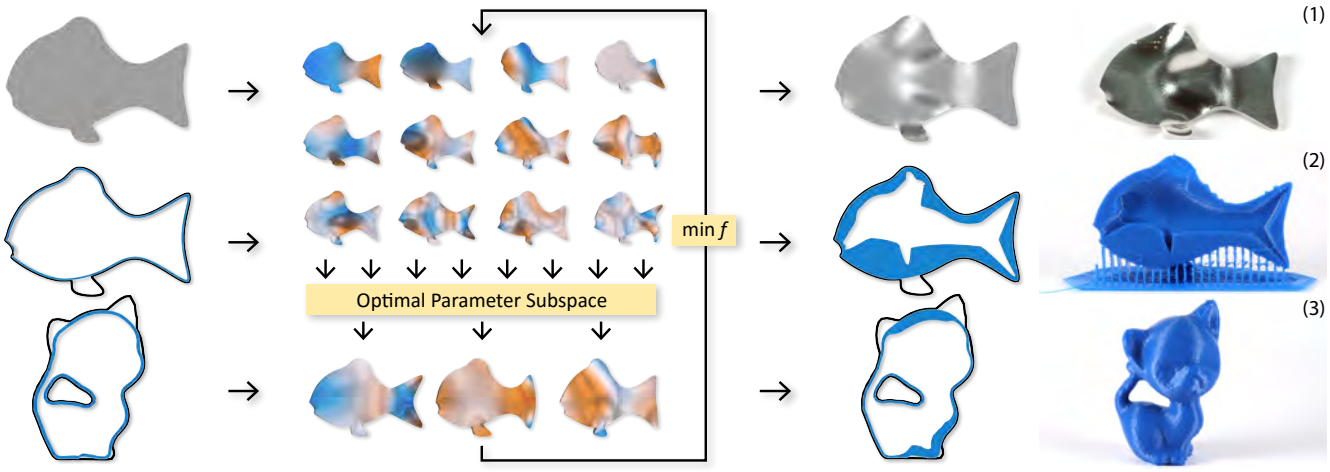


Figure 1: *We propose a novel method for local optimization that is applicable to various shape optimization problems, like (1) optimization of natural frequencies, (2) optimization of mass properties, and (3) optimization of the structural strength of solid objects. Our method computes locally optimal subspace projections that provide low-dimensional parameterizations. These are suitable for efficient local optimization.*

Abstract

In this paper we present a novel method for non-linear shape optimization of 3d objects given by their surface representation. Our method takes advantage of the fact that various shape properties of interest give rise to underdetermined design spaces implying the existence of many good solutions. Our algorithm exploits this by performing iterative projections of the problem to local subspaces where it can be solved much more efficiently using standard numerical routines. We demonstrate how this approach can be utilized for various shape optimization tasks using different shape parameterizations. In particular, we show how to efficiently optimize natural frequencies, mass properties, as well as the structural yield strength of a solid body. Our method is flexible, easy to implement, and very fast.

Keywords: geometry processing, shape optimization, geometric design optimization, non-convex optimization, digital fabrication

Concepts: •**Theory of computation** → **Computational geometry;** •**Mathematics of computing** → *Nonconvex optimization;* •**Computing methodologies** → *Mesh geometry models;*

*Joint first authors.

†Joint last authors.

SIGGRAPH '16 Technical Paper,, July 24 - 28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925886

1 Introduction

Shape optimization is a research field where computational tools for optimizing structures described by physical or mechanical models are developed. With digital fabrication becoming more and more a consumer-level technology, the demand for novel algorithmic shape optimization solutions is constantly growing. This is also reflected in a recent series of papers in the computer graphics community which aim at a broad range of diverse shape optimization problems, like structural stability of 3d printed objects [Stava et al. 2012; Lu et al. 2014; Zhu et al. 2012], objects with desired physical properties [Prévost et al. 2013; Bächer et al. 2014; Musialski et al. 2015], or, most recently, objects with desired natural frequency spectra [Bharaj et al. 2015]. All these solutions are rather specialized and aim at the computational design of some specific aspects of structure. However, what they have in common is that they optimize some specific shape properties parameterized in a well-chosen design space, and all rely on mathematical optimization of the given structure, where the objective and the constraints depend on the underlying shape and are in general highly non-convex.

In this paper, we introduce a novel shape optimization framework that provides a unified representation of these problems with respect to generic shape properties and parameterizations. Additionally, we provide an improved local minimization strategy based on the efficient exploration of an underdetermined shape design space, and we extend our method with soft box constraints, making it even more effective if only such constraints are present.

Our method aims at fast and robust non-convex local shape optimization and is inspired by the following observation: If the design space of a non-convex shape optimization problem is too small—particularly if it is (nearly) fully determined—it is difficult to find a solution at all, especially without the knowledge of appropriate initial values. Making the design space richer provides a remedy, since more local minima of the objective function exist, and a potentially good design can be found more easily from any starting position using local optimization. On the other hand, optimization in such high-dimensional and underdetermined spaces is slow and

difficult, especially if the evaluation of the gradients of the involved functionals is expensive. Indeed, since generic local optimization routines do not take any specific knowledge of the underdetermined space into account, they often take very long to reach a minimum or do not converge at all. However, our observation is that one can exercise local control over many independent shape properties, like natural vibration frequencies, or mass properties, by varying the shape parameters in a low-dimensional affine subspace.

Our method tries to exploit that fact: The shape is parameterized in a rich design space, however, our algorithm repetitively reduces this space to a locally best matching subspace that is fully determined. Best matching means that the design variables and the shape properties (e.g., natural frequencies) are decorrelated in the chosen local subspace. Subsequently, a local optimization (e.g., quasi-Newton or Levenberg–Marquardt) is performed in the subspace as long as the shape parameters are sufficiently decorrelated. When this is not the case anymore, the problem is again transformed to a new basis in the embedding design space. This procedure is repeated until convergence to a local minimum.

In summary, the contributions of this paper are the following:

- We provide a novel subspace parameterization for local gradient-based shape optimization problems that is more efficient than traditional generic methods. In particular, our method performs both faster and it converges more reliably to a valid local minimum.
- We introduce a constraint mapping strategy that allows us to formulate box constraints in the objective function. This turns out to be very efficient in combination with the subspace projection algorithm.
- We show how to apply our algorithm to a number of problems using different parameterizations. Specifically, we demonstrate how this approach can be used for (1) optimization of natural frequencies, (2) optimization of mass properties, and (3) optimization of structural strength of a solid object.

2 Related Work

Shape Optimization. Shape optimization aims at the automatic computation of structural or mechanical designs that suit some desired (global) goals [Delfour and Zolésio 2011]. It has been studied in the field of optimal control theory and structural engineering for quite some time, and it spans multiple scientific fields, like classical geometry, analytic geometry, functional analysis, partial differential equations, as well as (constrained) optimization, with applications to classical mechanics of continuous media such as elasticity theory.

In computer graphics, research on shape optimization became popular due to its applications to digital fabrication. Recent approaches provide algorithms for improving models for 3d printing, like the computation of structural stability [Stava et al. 2012], worst-case structural analysis [Zhou et al. 2013], cost-effective material usage [Wang et al. 2013], or optimization of both the strength and weight of printed objects [Lu et al. 2014]. Another series of papers deals with the physical mass properties of solids that optimize 3d objects to fulfill various objectives, like stable standing, rotating, or floating in a liquid [Prévost et al. 2013; Bächer et al. 2014; Musialski et al. 2015; Wang and Whiting 2016]. Recently, works that aim at the optimization of the natural frequency spectra of fabricated 3d objects in order to make them produce desired sounds have been published [Umetani et al. 2010; Bharaj et al. 2015; Hafner et al. 2015].

Computational Design. Shape optimization can also be seen as a subfield of general computational design problems, which are currently under very active research. Various methods have been proposed that integrate shape optimization into interactive modeling tools in order to solve specific problems. For instance, interactive systems for computational design tasks, like garment editing [Umetani et al. 2011], design of physically valid furniture [Umetani et al. 2012], articulated 3d-printed models [Bächer et al. 2012], elastically deformable objects [Skouras et al. 2013; Pérez et al. 2015], or construction of inflatable structures with desired shapes [Skouras et al. 2014] have been proposed.

Another example is material design, where the specification of a desired deformation behavior can be given *a priori*, and the composite material with matching elastic behavior is computed by optimization [Bickel et al. 2010]. Recently, a method for the computational design of elastic materials with constant physical structures has been proposed [Schumacher et al. 2015; Panetta et al. 2015].

Subset Methods. Modifications to the Levenberg-Marquardt algorithm that iteratively choose a subset of parameters have recently been proposed. The motivation is to increase numerical stability [Ipsen et al. 2011] and efficiency by removing parameters of low sensitivity and re-evaluating the full problem occasionally [Finsterle and Kowalsky 2011]. Our work has a similar goal, but generalizes the idea to arbitrarily oriented parameter subspaces and provides a criterion to decide when re-evaluation is necessary.

3 Shape Representation

First, we provide the definitions of all particular building blocks. Figure 2 provides an overview of the framework, where we define it here in a generic fashion, but we show in Section 6 how it can be applied to a number of specific shape optimization tasks.

Shape as a Variable. In the context of shape optimization methods, a shape is generally considered as a geometric domain $\Omega \subset \mathbb{R}^3$. In this paper, we represent a 3d shape by its boundary $\partial\Omega$, which is a 2-manifold surface, and we assume that $\partial\Omega$ encloses a bounded region of space, which is considered as the inside. It corresponds to the volume (and respectively the mass) of the object, where we assume a uniform mass density. In practice, we expect that $\partial\Omega$ is represented by a 2-manifold polygonal mesh with n vertex positions concatenated to the vector $\mathbf{x} \in \mathbb{R}^{3n}$.

Shape Parameterization. Let χ_0 be an initial geometric model represented by vertices $\mathbf{x}$ that is to be optimized. We assume that the shape optimization is achieved by controlling a set of *design variables* $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]$, i.e., the actual geometric variables $\mathbf{x}$ are the range of a function $\chi(\boldsymbol{\alpha})$ with $\chi_0 = \chi(\mathbf{0})$. Then $\chi(\boldsymbol{\alpha})$ is a (linear or non-linear) map encoding the 3d coordinates of all vertices:

$$\chi\colon \boldsymbol{\alpha} \in \mathcal{D} \subseteq \mathbb{R}^m \mapsto \mathbf{x} \in \mathbb{R}^{3n},$$

where $\mathcal{D}$ constitutes the *design space* which covers all admissible shapes. In the simplest case, χ can be an affine map

$$\chi(\boldsymbol{\alpha}) = \chi_0 + \mathbf{X}\,\boldsymbol{\alpha}, \tag{1}$$

where the columns of the $(3n \times m)$-matrix $\mathbf{X}$ play the role of discrete shape functions. Common choices for the shape functions would be radial bumps [Botsch and Kobbelt 2005] or manifold harmonics [Vallet and Lévy 2008]. In Section 5 we discuss further details of the used parameterizations.

Material Parameters. All shapes we optimize represent real objects and should be fabricable. Thus, the physical properties of the underlying material need to be provided. In cases where the shape property is derived from the equations of motion, the object is discretized using a finite element model with second-order elements [Bathe 2006]. The material is represented with an isotropic model, whose parameters are given by the Young's modulus E, Poisson's ratio ν, and the material density ρ.

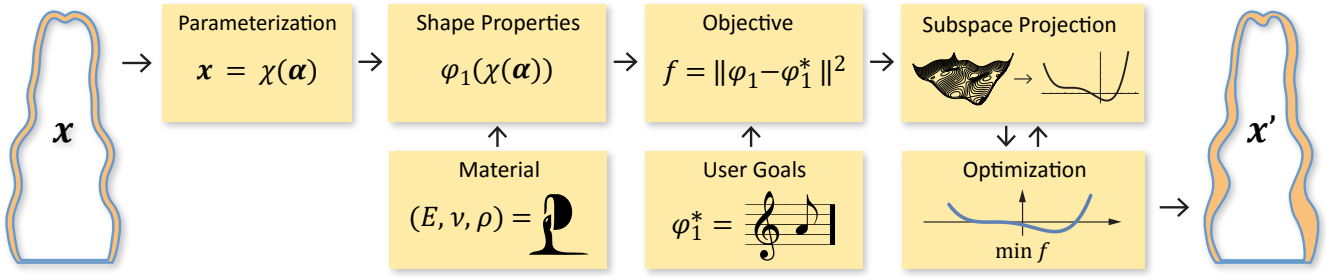


Figure 2: *Overview of our method on the example of optimizing the natural frequencies of a shape. Given a shape, represented by its geometric variables* $\mathbf{x}$*, first we provide a parameterization* $\boldsymbol{\chi}$*. Further, the shape properties can be computed using the parameterization and some given material properties. Next, given user desired values for the properties, we perform a subspace projection and optimize the shape in order to fulfill these goals. Finally, the output is the optimized shape* $\mathbf{x}'$*.*

Shape Properties. The aim of shape optimization is to find an optimal shape, i.e., one which meets some geometric requirements and at the same time exhibits some desired properties. These entities are further denoted as the so-called *shape property variables*, collected to the vector-valued function $\boldsymbol{\varphi}(\boldsymbol{\chi}) = [\varphi_1(\boldsymbol{\chi}), \ldots, \varphi_k(\boldsymbol{\chi})]$. It delivers the properties of the shape with respect to the design variables $\boldsymbol{\alpha}$, and it is usually a non-linear and non-convex mapping. Often, especially in structural and mechanical problems, shape properties are given as solutions to a PDE that governs the problem. As we will see later in Section 6, a component $\varphi_i(\boldsymbol{\chi})$ can stand for, e.g., a geometric property, a mass property, the frequency of an eigenmode, or any other feature of the object.

User Goals. The final ingredients to our system are the actually desired target values φ_i^* for the shape properties. In particular, if the goal is to optimize the natural frequency spectrum, the target needs to be a set of values for the particular fundamental tone (i.e., the pitch) and the overtones (cf. Section 6.1). In other cases, like mass properties, the actual behavior of the object if exposed to forces needs to be specified (cf. Section 6.2). Usually, such desired values are given by the user. The shape optimization is successful if these shape properties $\varphi_i(\boldsymbol{\chi})$ take on the desired target values φ_i^*.

4 Shape Optimization

Our goal is to provide a generic methodology which allows creating shapes with desired properties independent of the chosen parameterization. In this section we propose a novel local optimization scheme for efficient shape optimization.

4.1 Shape Optimization Setup

Since we couple the property variables to the design variables via the geometry of the shape, a setup for a formal optimization problem can be stated in *nested form* with the objective functional f defined with respect to the property values $\boldsymbol{\varphi}$ of the design variables $\boldsymbol{\alpha}$ as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} f(\boldsymbol{\varphi}(\boldsymbol{\chi}(\boldsymbol{\alpha}))) \\ &\text{s.t. } g_j(\boldsymbol{\chi}(\boldsymbol{\alpha}), \boldsymbol{\varphi}(\boldsymbol{\chi}(\boldsymbol{\alpha}))) \leqslant 0\,. \end{aligned} \tag{2}$$

Besides the properties of a shape, the optimization usually has to take a number of (quality or inequality) constraints $g_j(\boldsymbol{\chi}, \boldsymbol{\varphi}(\boldsymbol{\chi})) \leqslant 0$ into account, which can either depend directly on the shape and/or on its property functions. In most cases, the constraints are necessary in order to enforce physical plausibility of the shapes, however, in Section 5.3 we propose a method to reformulate some of them as part of the objective function.

In this paper, the actual scalar-valued objective function f is a (weighted) sum of squared differences between the property variables and the given target values. Therefore, the generic formulation of our shape optimization objective is

$$f(\boldsymbol{\alpha}) = \sum_{i=1}^{k} \omega_i \, \|\varphi(\boldsymbol{\chi}(\boldsymbol{\alpha}))_i - \varphi_i^*\|^2 \,,$$

where ω_i are problem specific weighting factors. We will see in the application section that many practically relevant application cases turn out to be special instances of this generic optimization setting.

4.2 Sensitivity Analysis

If the objective function f, the property function $\boldsymbol{\varphi}$, and the parameterization $\boldsymbol{\chi}$ are smooth and differentiable with respect to $\boldsymbol{\alpha}$, we can use an iterative gradient-based optimization algorithm to obtain a local minimum of the function f. While current numerical optimization routines are able to compute the derivatives numerically (e.g., using finite differences), better results can be obtained if analytical gradient functions can be supplied [Nocedal and Wright 2006]. Due to the nested form of the objective function, the gradient can be expressed using the chain rule as:

$$\nabla_{\boldsymbol{\alpha}} f = \frac{\partial f}{\partial \boldsymbol{\varphi}} \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \,. \tag{3}$$

Additionally, in the case of (non-linear) constrained optimization, we also need to provide the gradients of g_i:

$$\nabla_{\boldsymbol{\chi}} g_i = \frac{\partial g_i}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \quad \text{and} \quad \nabla_{\boldsymbol{\varphi}} g_i = \frac{\partial g_i}{\partial \boldsymbol{\varphi}} \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \,. \tag{4}$$

In general, equipped with the ingredients from Equations (2) to (4), we could perform local optimization using standard optimization routines.

4.3 Subspace Projection

Rationale. Applying standard local numerical solvers to the generic optimization problem (2) usually leads to very poor convergence. This is due to the fact that the m (typically in the range of 10s to 100s) shape parameters $\boldsymbol{\alpha}$ make the problem heavily underdetermined, which creates many redundant (shallow) directions where shape changes do not imply significant changes in the target functional. Moreover, since each shape parameter α_i can significantly influence several (or even all) shape properties $\varphi_i(\boldsymbol{\chi})$ in a non-linear fashion, there are potentially many local minima in which iterative solvers can get trapped. Finally, since (at least) the gradient of the target functional needs to be evaluated in each iteration, the large number of shape parameters negatively affects the compute cost per iteration.

One option would be to reduce the dimensionality of the design space $\mathcal{D}$. However, if the design space is too restricted, i.e., too

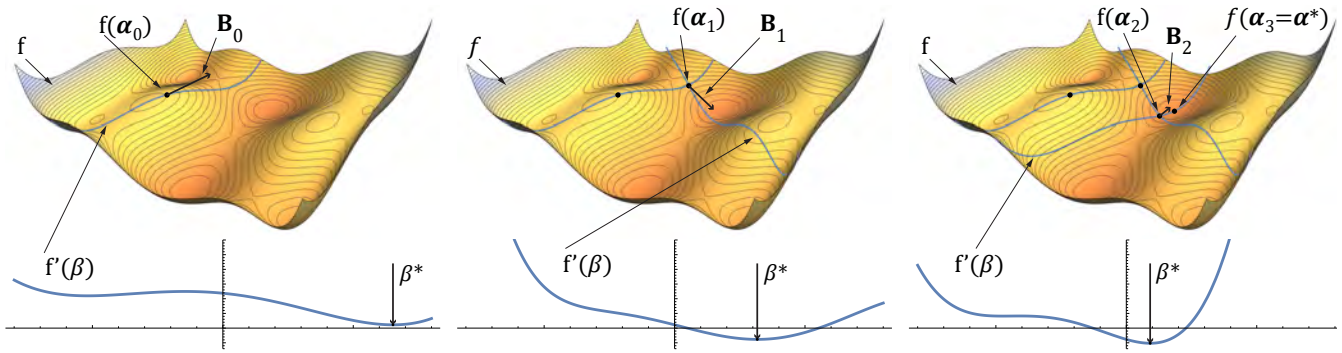


Figure 3: *Iterative subspace projection depicted illustratively on a map $\mathbb{R}^2 \mapsto \mathbb{R}$. Starting with an initial value $\boldsymbol{\alpha}_0$, our method computes an optimal subspace $\boldsymbol{\alpha}_0 = \mathbf{B}_0\boldsymbol{\beta}$, depicted as the vector $\mathbf{B}_0$, and the optimization is performed using a line-search method in the subspace until a minimum β^* it found, or the subspace does not decorrelate the variables anymore (left). Then, a new subspace is computed and the procedure is repeated (middle), until a local minimum $\boldsymbol{\alpha}^*$ is found (right). Please note that for the purpose of visualization we assume that the particular property function $\boldsymbol{\varphi}$ is at the same time the actual objective f, which is not the case in general. Moreover, for illustration purposes, we run the inner local optimization until it finds a local minimum $\boldsymbol{\beta}^*$ in $\mathcal{R}$.*

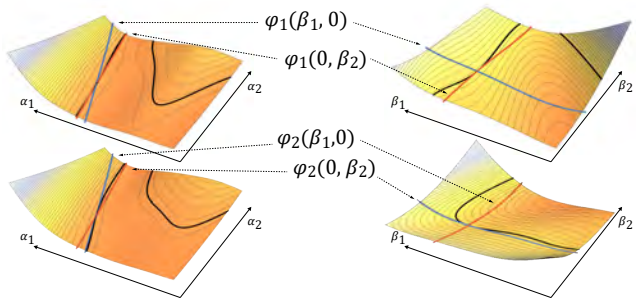


Figure 4: *Illustration of the shape property decorrelation on an $\mathbb{R}^2 \mapsto \mathbb{R}^2$ example. Left: two property functions $\varphi_1(\alpha_1, \alpha_2)$ and $\varphi_2(\alpha_1, \alpha_2)$ are mapped to $\varphi_1(\beta_1, \beta_2)$ and $\varphi_2(\beta_1, \beta_2)$.*

low-dimensional or even fully determined, a solution could be non-existent or very difficult to find with a local optimization approach without appropriate initial values. On the other hand, our key observation is that shape properties, while non-linear, can locally be controlled in a low-dimensional affine parameter subspace. This is achieved by decorrelating the shape properties, even if they originally exhibit strong correlations in the embedding design space (cf. Fig. 4).

Local Preconditioning. We hence propose a numerical algorithm that is based on a proper (local) pre-conditioning of the problem such that negative effects emerging from underdetermined-ness and interference between shape parameters are lessened or even eliminated. Intuitively, the idea is to find a projection from the design space $\mathcal{D}$ into a local subspace $\mathcal{R} \subseteq \mathbb{R}^k$ with new shape parameters $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_k] \in \mathcal{R}$, each dominantly controlling one of the shape properties $\varphi_i(\boldsymbol{\chi})$ and having only little influence on the others. Due to the non-linearity of the shape properties, finding such a reparameterization globally would be very challenging. However, if we restrict ourselves to a local region in the design space, i.e., the vicinity of the current location $\boldsymbol{\alpha}$, a linear map

$$\boldsymbol{\alpha} = \mathbf{B}\,\boldsymbol{\beta}$$

with an $(m \times k)$ parameter transform matrix $\mathbf{B}$ turns out to be sufficient. Figure 4 depicts the effect of the decorrelation of the property functions in the subspace $\mathcal{R}$, while their projections into the design space $\mathcal{D}$ exhibit strong correlations.

The (local) dependency of the shape properties $\boldsymbol{\varphi}$ on the new shape parameters $\boldsymbol{\beta}$ is expressed by the gradient

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \mathbf{X}\,\mathbf{B}$$

where, for simplicity, we assume $\boldsymbol{\chi}$ to lie in an affine space spanned by the matrix $\mathbf{X}$ (cf. Eq. (1)). At a given design space location $\boldsymbol{\alpha}$ (or, equivalently, for a given intermediate shape $\boldsymbol{\chi}(\boldsymbol{\alpha})$) we obtain maximum decorrelation between the new shape parameters $\boldsymbol{\beta}$ if

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \mathbf{X}\,\mathbf{B} = \mathrm{diag}(\gamma_1, \ldots, \gamma_k).$$

Here, the diagonal coefficients $\gamma_i = \omega_i \varphi_i^*$ are chosen such that the scale differences between different shape properties φ_i are properly compensated. Setting $\gamma_i = 1$ would lead to potentially large anisotropies if, e.g., the different shape properties are measured in different physical units. The matrix $\mathbf{B}$ can easily be computed as the least-norm solution of

$$\min \left\| \tilde{\mathbf{B}} \right\|_2^2 \quad \text{s.t.} \quad \left[\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \right] \tilde{\mathbf{B}} = \mathbf{I}_k\,, \tag{5}$$

where $\mathbf{B} = \tilde{\mathbf{B}}\,\mathrm{diag}(\gamma_1, \ldots, \gamma_k)$, and $\mathbf{I}_k$ is a $k \times k$ identity matrix. Choosing the least-norm solution preserves the design space structure in the sense that each parameter β_i controls the property function φ_i mostly through those α_j that also have a strong influence on φ_i.

Algorithm. Our iterative numerical optimization scheme runs two nested loops. In the outer loop, the shape parameter transform matrix $\mathbf{B}$ is computed for the current design space location $\boldsymbol{\alpha}$ (or, equivalently, for the current intermediate shape $\boldsymbol{\chi}(\boldsymbol{\alpha})$). In the inner loop, gradient-based iterations are applied to optimize the new (local) shape parameters $\boldsymbol{\beta}$. The inner loop stops if either a local minimum is found or if the diagonal dominance

$$\frac{1}{k} \sum_{i=1}^{k} \frac{|G_{ii}|}{\sum_j |G_{ij}|}$$

of the gradient matrix

$$\mathbf{G} = [G_{ij}] = \left[\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\chi}} \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\alpha}} \right] \mathbf{B} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\beta}} \tag{6}$$

falls below a certain threshold τ, which is usually chosen in the range of $0.5 < \tau < 1.0$. The latter indicates that the shape parameters $\boldsymbol{\beta}$ are no longer sufficiently decorrelated. In this case, the next

outer iteration starts with the computation of a new matrix $\mathbf{B}$ at the updated design space position $\boldsymbol{\alpha}(\boldsymbol{\beta}) = \mathbf{B}\boldsymbol{\beta}$.

In each local optimization step of the inner loop, the gradient $\mathbf{G}$ only depends on $k \ll m$ parameters. This means that the derivatives of $\boldsymbol{\varphi}$ need only be computed w.r.t. the parameters $\boldsymbol{\beta}$ of the subspace. This constitutes a significant reduction in computational cost compared to the evaluation of the gradient in the full design space. The inner loop continues as long as the subspace parameters $\boldsymbol{\beta}$ retain effective control over the shape properties, which is expressed by the diagonal dominance criterion above.

Our method turns out to be very efficient for optimization of shapes parameterized in an underdetermined design space. As we show later in the results (cf. Section 6), our algorithm performs both faster and it converges more reliably to a good minimum compared to a standard local optimization. Figure 3 shows a simplified summary of the method in a low-dimensional case ($\mathcal{D} \in \mathbb{R}^2 \mapsto \mathcal{R} \in \mathbb{R}$).

4.4 Choice of the Local Method

The shape optimization algorithm using iterative subspace projection employs continuous local optimization in the inner loop. The choice of the local optimization algorithm depends on the application, the objective function, and the constraints of problem. Recent works in shape optimization favor BFGS-type algorithms like quasi-Newton (QN) and the sequential quadratic programming algorithm (SQP), which is why we focus our evaluation on this class. Their great advantage is that box constraints, linear constraints, and non-linear constraints, which are ubiquitous in many shape optimization tasks, are supported by off-the-shelf implementations like the ones provided in MATLAB.

If the nature of the problem facilitates the constraint mapping strategy discussed in detail in Section 5.3 and contains no additional constraint beyond that, we show how it can be transformed into an unconstrained problem. In this case, the special non-linear least-squares structure of the objective function permits the use of the Levenberg-Marquardt (LM) algorithm as an alternative to BFGS. We conduct an evaluation of both QN and LM using subspace projection in Section 6.1. The results reveal that subspace projection accelerates both algorithms significantly and that the performance of LM is superior to that of QN (cf. Figure 9 and Table 1).

5 Parameterizations and Constraints

This section describes the parameterizations used in the applications of Section 6. We also introduce a constraint mapping scheme that relaxes the restrictions imposed on the design space by box constraints and improves the effectiveness of our subspace projection method.

5.1 Parameterizations

Manifold Harmonics. For most of our applications, we use a parameterization given by the manifold harmonics of the surface of the shape, as recently proposed by Musialski et al. [2015], which we briefly summarize in this section. It defines a 3d volume as the volume enclosed by two 2-manifold surfaces, the outer surface $\overline{\mathcal{M}}$ and the inner surface $\underline{\mathcal{M}}$. If the surfaces have boundaries, the closed 3d model is formed by connecting them.

The outer surface $\overline{\mathcal{M}}$ is given by the user and remains constant throughout the optimization in order not to distort the appearance of the model. The inner surface $\underline{\mathcal{M}}$ is defined as an offset surface to $\overline{\mathcal{M}}$, where the per-vertex offset directions $\boldsymbol{\nu}_i \in \mathcal{V}$ are precalculated and the per-vertex offset magnitudes δ_i are controlled by the optimization variables. For the vector field $\mathcal{V}$ we utilize the surface normals (in Section 6.1), or a mean curvature flow vector field [Tagliasacchi et al. 2012] (in Sections 6.2 and 6.3).

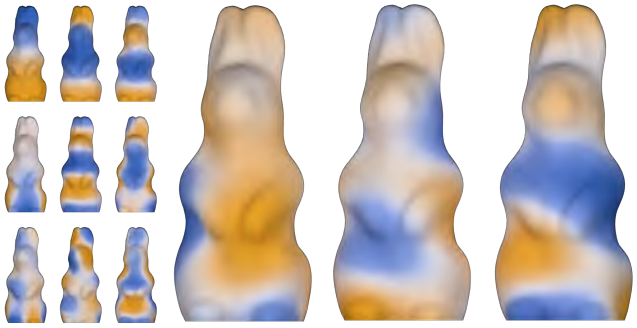


Figure 5: *Left: Nine of the first 24 manifold harmonics on the rabbit mesh. Right: Influence regions of the parameters in the reduced design space for the optimization of natural frequencies.*

Since the surfaces are given as triangle meshes, this would usually introduce an unacceptably large number of design parameters $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_n]^T$, which is equal to the number of vertices. In order to deal with this problem, we utilize the manifold harmonics parameterization. It uses the k eigenvectors corresponding to the k smallest eigenvalues of the discrete Laplace-Beltrami operator on $\overline{\mathcal{M}}$. These vectors constitute an orthonormal basis, often denoted as manifold harmonic basis [Vallet and Lévy 2008], which can be used to build a set of design parameters $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \in \mathcal{D}$, where $m \ll n$. Let $\boldsymbol{\Gamma}_m = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_m]$ be the matrix containing these m eigenvectors in its columns. Then a vertex $\underline{\mathbf{x}}_i$ of the inner surface can be expressed through an offset to the corresponding vertex $\overline{\mathbf{x}}_i$ of the outer surface as (cf. Eq. (1))

$$\underline{\mathbf{x}}_i = \overline{\mathbf{x}}_i + \delta_i \boldsymbol{\nu}_i, \quad \boldsymbol{\delta} = \boldsymbol{\Gamma}_m \boldsymbol{\alpha}. \tag{7}$$

Here, the offsets are spanned by the basis $\boldsymbol{\Gamma}_m$, and the coefficients are the design parameters $\boldsymbol{\alpha}$. Fig. 6 (left) portrays the influence of the individual design parameters α_i in two dimensions.

It is important to enforce geometric constraints that prevent self-intersections of $\underline{\mathcal{M}}$ and intersections between $\overline{\mathcal{M}}$ and $\underline{\mathcal{M}}$, as illustrated in Fig. 6. Local self-intersections of $\underline{\mathcal{M}}$ can be reduced through a good choice of offset directions $\boldsymbol{\nu}_i$. To prevent global intersections, box constraints on the offset magnitudes δ_i have to be introduced. Using the definition in Eq. (7), they can be written as

$$\boldsymbol{\delta}_l \preceq \boldsymbol{\Gamma}_m \boldsymbol{\alpha} \preceq \boldsymbol{\delta}_u. \tag{8}$$

The lower bounds $\boldsymbol{\delta}_l$ can be used to enforce a minimum wall thickness, i.e., a minimum distance between $\overline{\mathcal{M}}$ and $\underline{\mathcal{M}}$. However, due to the global support of Laplacian eigenvectors, these box constraints can greatly diminish the freedom of all design parameters $\boldsymbol{\alpha}$. This happens if the lower and upper offset bounds are in close proximity even for small regions of the model. The constraint mapping scheme in Section 5.3 proposes a remedy for this problem.

Cage Deformation. In Section 6.1, we show that optimization using subspace projection can also be combined with cage parameterizations, which are used by Bharaj et al. [2015] for the optimization of frequency spectra. A 2d shape is embedded into a deformation cage with n control points (cf. Figure 10). Three of the $2n$ degrees of freedoms are fixed in order to eliminate rigid-body modes, yielding a $(2n-3)$-dimensional design space $\mathcal{D}$. In order to create a 3d solid, the deformed 2d shape is extruded along its normal by a constant thickness value.

Symmetric Bell Parameterization. Radially symmetric instruments, like church bells and handbells, are best optimized using a parameterization that preserves the symmetry. Otherwise, orthogonal pairs of mode shapes with the same frequency, which are intrinsic to radially symmetric shapes, will be lost. The parameterization

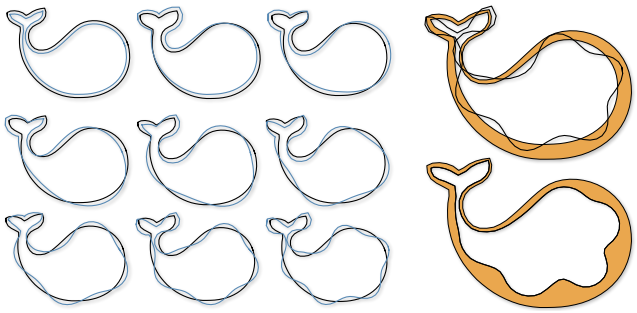


Figure 6: *Left: Manifold harmonics with increasing frequencies. Top right: Perturbations in the design variables easily violate constraints in thin regions of the model. Bottom right: Constraint mapping performs a soft clamping of offset magnitudes near the bounds.*

we use is based on variations of the wall thickness in the model's cross section. Every parameter has local support and thickens or thins the wall at a particular location. A result using this parameterization is shown in Section 6.1.

5.2 Choice of the Parameterization

The shape parameterizations discussed in Section 5.1 each have their merits in terms of geometric faithfulness to the target shape, ease of optimization, and available manufacturing methods. Most of the results in this paper use a manifold harmonics basis to parameterize the shape space. One advantage is that the silhouette of a 2d shape or the outer surface of a thin-shell model can easily be preserved this way, while the desired shape properties are attained through thickness modifications. Secondly, the constraint mapping strategy can be applied to eliminate box constraints, which leads to an unconstrained optimization problem in many cases.

Cage-based deformation of a 2d shape has the advantage that it permits the use of cheaper fabrication techniques like laser cutting an object from sheet metal. However, this parameterization is less geometrically faithful because it allows large deviations from the original silhouette. Additionally, it is necessary to introduce constraints on the cage handles to avoid non-physical deformations like local inversions of the silhouette.

The special-purpose parameterization for radially symmetric objects preserves orthogonal shape modes by construction. Therefore, the number of target modes can be reduced to simplify optimization because coupled modes will automatically retain the same frequency. Results of this parameterization can often be manufactured on a CNC lathe.

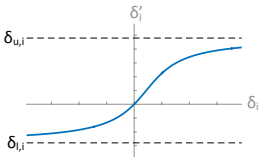
5.3 Constraint Elimination

As mentioned, box constraints like those in Eq. (8) impose strong restrictions on the design space. We therefore introduce a constraint-mapping scheme to eliminate them, which offers a number of advantages:

- It counteracts the degradation of the design space that is often introduced by box constraints. This is achieved by modifying the shape parameterization to obey the constraints by construction.
- It improves the effectiveness of the subspace projection algorithm described in Section 4.3.
- In applications that have no additional hard constraints, it enables the usage of unconstrained optimization routines like Levenberg–Marquardt or quasi-Newton, which provide runtime advantages over constrained optimization routines.

Our constraint mapping scheme defines a transfer function for each per-vertex offset magnitude δ_i to eliminate a constraint of the form $\delta_{l,i} < \delta_i < \delta_{u,i}$. The mapped offset δ_i' is calculated via a "soft" clamping of δ_i to the bounds $\delta_{l,i}$ and $\delta_{u,i}$ via

$$\delta_i' = \frac{\delta_{u,i} - \delta_{l,i}}{\pi} \tan^{-1}(\delta_i - o_i) + o_i, \text{ with}$$
$$o_i = \frac{\delta_{u,i} + \delta_{l,i}}{2}.$$



The graph of the mapping is plotted for a particular $\delta_{l,i}$ and $\delta_{u,i}$ in the figure to the left. Note that at the beginning of an iteration, the origin of each vertex offset δ_i is additionally shifted such that the constraint mapping function maps $\delta_i = 0$ to $\delta_i' = 0$. Therefore, $\delta_i = 0$ corresponds to the initial position of the inner surface $\underline{\mathcal{M}}$ at that iteration. The original shape parameterization from Eq. 7 is replaced by

$$\underline{\mathbf{x}}_i = \overline{\mathbf{x}}_i + \delta_i' \boldsymbol{\nu}_i. \tag{9}$$

Since δ_i' is ensured to be in the bounded region, the box constraints can be removed. Fig. 6 (right) shows how constraint violations induced by a perturbation in a high-frequency component are solved by constraint mapping.

Although this is not a novel method to remove constraints, it has a particular advantage in combination with the subspace projection algorithm. Eq. (5) uses the gradients $\partial\boldsymbol{\varphi}/\partial\boldsymbol{\alpha}$ of the property variables with respect to the shape parameters. Comparing the two versions of the gradients,

$$\frac{\partial\boldsymbol{\varphi}}{\partial\boldsymbol{\delta}}\frac{\partial\boldsymbol{\delta}}{\partial\boldsymbol{\alpha}} \quad \text{and} \quad \frac{\partial\boldsymbol{\varphi}}{\partial\boldsymbol{\delta}'}\frac{\partial\boldsymbol{\delta}'}{\partial\boldsymbol{\delta}}\frac{\partial\boldsymbol{\delta}}{\partial\boldsymbol{\alpha}},$$

with and without constraint mapping, the only difference is the additional factor corresponding to the slope of the transfer function. As is evident in the inset figure above, the slope has low values near the bounds, and higher values in the region far away from the bounds.

Consequently, the value of $\partial\varphi_i/\partial\alpha_j$ is lowered if the shape regions controlled by α_j are close to the bounds. Since Eq. (5) is a least-norm problem, such parameters will barely contribute to the reduced design space $\mathcal{R}$ if there are suitable parameters that control regions further away from the bounds. The effect is that regions of $\underline{\mathcal{M}}$ which have been moved close to the bounds already in a previous iteration are less likely to be picked up again in the current iteration. Furthermore, the optimization routine concentrates on regions further away from the bounds, which are less restricted in movement and thus have a higher potential to decrease the objective function.

6 Applications and Results

Many objectives of shape optimization applications can be written as a (weighted) sum of property variables or powers thereof. We have implemented three shape optimization problems, all of which are defined in terms of properties and use the subspace projection algorithm. Specifically, the effectiveness of subspace projection is shown on the examples of frequency spectrum optimization, the optimization of mass properties for static stability and buoyancy, and the optimization of structural strength under given load conditions.

6.1 Optimization of Natural Frequencies

The natural spectrum of a 3d object describes the physically realizable oscillations that the object can undergo in the absence of external loads. A particular manifestation is the lingering sound of a musical instrument after it has been struck by the player. The sound depends on both material and shape of the object and can be decomposed into a sequence of tones that correspond to the oscillatory frequencies. The challenge of frequency optimization is

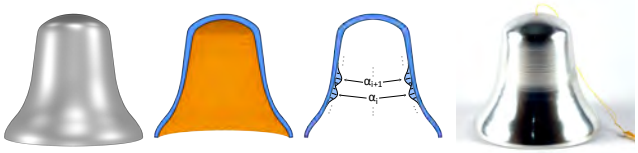


Figure 7: *From left to right: Rendering of outer surface; Cross section of optimized model; Influence of parameters; Photo of milled result.*

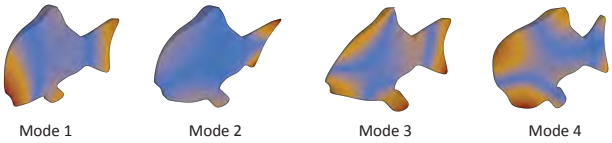


Figure 8: *The first four shape modes of a fish model, where the first corresponds to the frequency of the fundamental tone, and the following three to the overtones.*

to find a 3d model that is similar to a target shape and exhibits a set of target frequencies when fabricated from the target material. We show that our optimization algorithm is capable of solving this task by computing and fabricating a series of metallophones using different parameterizations.

Recently, this problem has also been tackled by Bharaj et al. [2015] using a combination of global and local optimization. We demonstrate that the local optimization of natural frequencies can be significantly accelerated by the subspace projection algorithm described in Section 4.3.

Objective. The natural frequencies of a non-trivial elastic solid cannot be determined analytically, and thus they are approximated with a finite-element model. To aid computation, the internal friction of the object is neglected in favor of an undamped system, which yields a generalized eigenvalue problem [Bathe 2006]

$$\mathbf{K}\boldsymbol{\nu}_i = \lambda_i \mathbf{M}\boldsymbol{\nu}_i, \quad \boldsymbol{\nu}_i^T \mathbf{M}\boldsymbol{\nu}_i = 1\,. \tag{10}$$

The stiffness matrix $\mathbf{K}$ and the mass matrix $\mathbf{M}$ are large, sparse matrices that depend on the finite-element discretization of the object. Every eigenpair $(\lambda_i, \boldsymbol{\nu}_i)$ with $\lambda_i > 0$ corresponds to an oscillation mode of the solid. The eigenvector $\boldsymbol{\nu}_i$ is referred to as the *shape mode* and describes the geometric deformation induced by the mode (cf. Figure 8). The frequency and amplitude of the mode are given by

$$\phi_i = \frac{1}{2\pi}\sqrt{\lambda_i} \quad \text{and} \quad a_i = \mathbf{F}^T |\boldsymbol{\nu}_i|,$$

where $\mathbf{F}$ is the force profile of the initial excitation, and $|\cdot|$ denotes the element-wise absolute value. The primary goal is the optimization of the frequencies ϕ_i, and therefore they are used as property variables $\varphi_i = \phi_i$. The objective is given by

$$f(\boldsymbol{\alpha}) = \sum_{i \in \mathcal{I}} \left\| \frac{\phi_i(\boldsymbol{\alpha}) - \phi_i^*}{\phi_i^*} \right\|^2, \tag{11}$$

where $\mathcal{I}$ is the index set of target modes, and ϕ_i^* are the target frequencies. As noted by Bharaj et al. [2015], the amplitudes can be optimized as well using a second optimization step

$$f_a(\boldsymbol{\alpha}) = \sum_{i \in \mathcal{I}} \| a_i(\boldsymbol{\alpha}) - a_i^* \|^2 \quad \text{s.t.} \quad \phi_i(\boldsymbol{\alpha}) = \phi_i^*, \ \ \forall i \in \mathcal{I},$$

however, we found this to have limited effect on the quality of the results.

Note that the lowest frequency ϕ_1 corresponds to the fundamental tone of the sound spectrum and determines its pitch. The higher frequencies $\phi_2, \phi_3, \ldots$ determine the overtones of the spectrum and thus the timbre of the sound.

Gradients. The subspace projection algorithm requires the derivatives $\partial\boldsymbol{\varphi}/\partial\boldsymbol{\alpha}$ in order to find an optimal subspace $\mathcal{R}$. These are given by the frequency derivatives

$$\frac{\partial \phi_i}{\partial \alpha_j} = \frac{1}{4\pi\sqrt{\lambda_i}} \frac{\partial \lambda_i}{\partial \alpha_j}, \quad \text{with} \quad \frac{\partial \lambda_i}{\partial \alpha_j} = \boldsymbol{\nu}_i^T \left(\frac{\partial \mathbf{K}}{\partial \alpha_j} - \lambda_i \frac{\partial \mathbf{M}}{\partial \alpha_j} \right) \boldsymbol{\nu}_i\,.$$

If the amplitudes are part of the objective, the derivatives of the eigenvectors are needed as well. In this case we combine the computation of the eigenvalue and eigenvector derivatives by solving the linear system

$$\begin{pmatrix} \mathbf{K} - \lambda_i \mathbf{M} & -\mathbf{M}\boldsymbol{\nu}_i \\ 2\boldsymbol{\nu}_i^T \mathbf{M} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \boldsymbol{\nu}_i}{\partial \alpha_j} \\ \frac{\partial \lambda_i}{\partial \alpha_j} \end{pmatrix} = \begin{pmatrix} \left(\lambda_i \frac{\partial \mathbf{M}}{\partial \alpha_j} - \frac{\partial \mathbf{K}}{\partial \alpha_j} \right) \boldsymbol{\nu}_i \\ -\boldsymbol{\nu}_i^T \frac{\partial \mathbf{M}}{\partial \alpha_j} \boldsymbol{\nu}_i \end{pmatrix}.$$

Please consult the supplemental document for a derivation of the analytical gradients of $\mathbf{K}$ and $\mathbf{M}$ for isoparametric elements. Computing these derivatives in a high-dimensional design space is a computationally expensive task and demonstrates a strength of the subspace projection method: We only evaluate the derivatives with respect to all design variables $\boldsymbol{\alpha}$ once in each subspace projection step. The local optimization routine on the other hand runs on a strongly reduced set of design variables $\boldsymbol{\beta}$, and therefore the gradient computation for local methods is a lot faster.

Parameterizations. For most of our results, a manifold harmonics basis as discussed in Section 5.1 is used to define the design space $\mathcal{D}$. This parameterization is suitable for both 2.5d shapes, like the glockenspiel in Fig. 9, and full 3d shapes, like the rabbit-shaped bell in Fig. 15. The dimension of the design space $\mathcal{D}$ was chosen to be 64 for the glockenspiel animals shown in Fig. 9 and the bell shown in Fig. 15. The dimension of the reduced design space $\mathcal{R}$, which equals the number of simultaneously optimized natural frequencies, varies between 3 and 4. Fig. 5 shows the influence regions of a few manifold harmonics parameters α_i with increasing frequencies, and the influence regions of the parameters $\beta_1, \beta_2, \beta_3$, which locally control the three target frequencies in a decorrelated manner.

Subspace projection can also be used in combination with cage-based deformation. The cage we use has nine control points, yielding a 15-dimensional design space $\mathcal{D}$ after eliminating rigid-body modes. For the optimization of three frequencies, the reduced design space $\mathcal{R}$ is controlled by three parameters $\beta_1, \beta_2, \beta_3$. Fig. 10 shows the influence of these parameters on the control points in order to optimize a fish-shaped metallophone.

For the optimization of thin-walled, radially symmetric shapes like church bells and handbells, we use the symmetric bell parameterization described in Section 5.1. We do not evaluate this parameterization in detail, but we have optimized a handbell to the tone A7 (3520 Hz) and produced it using a milling machine. The result is shown in Fig. 7 and demonstrated in the supplemental video.

Results. In order to show that optimization using subspace projection converges well regardless of the initial parameter values, we optimized the fish glockenspiel bar with 10 randomly chosen starting values, each one with and without subspace projection. Fig. 13 shows the objective value plotted against run time for all 20 iterations. Not only is the optimization using space projection almost three times as fast on average, it also converges to a valid solution for all initial parameter values. In comparison, only 8 of the 10 iterations converge when using the full design space. Table 2 lists statistics of the converged iterations.

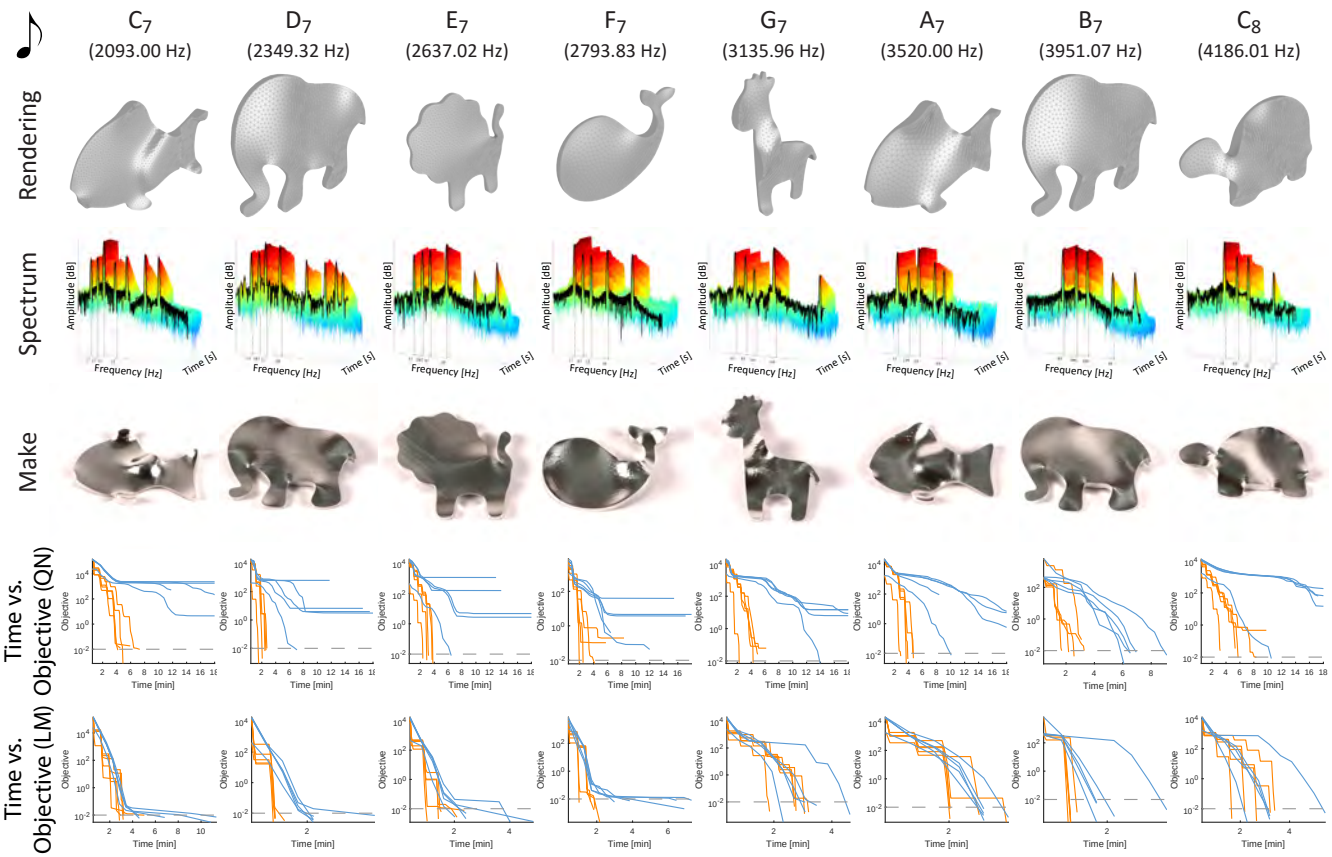


Figure 9: *Glockenspiel of 2.5d animals optimized using our method, where we optimized the fundamental and 3 overtones for each shape. The columns correspond to the desired pitch of each object. Row 1: resulting 3d models. Row 2: the measured frequency spectrum. Row 3: the fabricated instruments. Row 4/5: plots that track the decrease in objective value over time. Each shape was optimized with 5 initial values, once with subspace projection (orange), and once without (blue). Row 4: quasi-Newton solver. Row 5: Levenberg-Marquardt solver.*

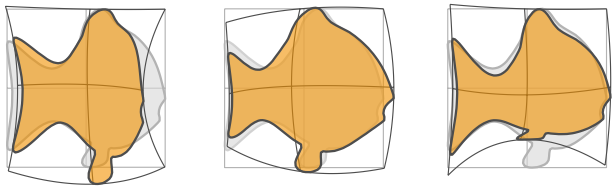


Figure 10: *Influence of the three parameters in the reduced design space on the cage handles.*

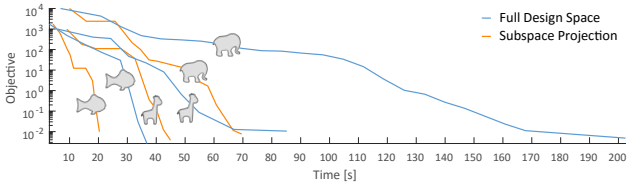


Figure 11: *Graph of run time versus objective values for the cage-based optimization of three animal models. The blue curves correspond to optimization on the full design space, the orange curves use subspace projection.*

As depicted in Fig. 9, we have optimized and produced a glockenspiel with bars in the shape of animals. This result is inspired by the work of Bharaj et al. [2015], who have optimized a similar instrument using cage-based deformation. In contrast, we use a manifold harmonics parameterization that varies the thickness of the bars while keeping the outlines of the input shapes intact to avoid strong distortions. As is evident from the graphs tracking the improvement of the objective value over time, our method (orange) converges more quickly on average and for a larger number of initial parameter values than optimization on the full design space (blue). Our constraint mapping scheme leads to an unconstrained optimization problem with a non-linear least-squares objective. This permits the use of the Levenberg-Marquardt (LM) algorithm, whose runtime performance we compare to a quasi-Newton (QN) solver. LM performs better than QN, both in terms of runtime and convergence. However, subspace projection improves performance for both optimization routines. Exact timings and the number of objective function evaluations are listed in Table 1.

Fig. 14 demonstrates the effect of changing the number of parameters in $\mathcal{D}$, and the number of frequencies that are being optimized. The blue bars show that the run time of the subspace projection algorithm only increases slightly if the number of parameters is increased, while optimization using the full design space becomes noticeably slower. Additionally, subspace projection converges for significantly more of the difficult test cases in which the number of parameters is low and the number of optimized frequencies is high.

We also evaluate subspace projection in combination with cage-based deformation. In Fig. 12, the difference between the original fish model and the optimized versions using cage deformation and manifold harmonics is demonstrated. The manifold harmonics version has a geometry which is much closer to the original because

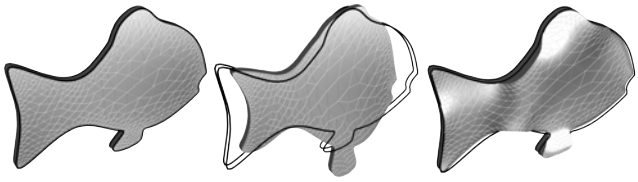


Figure 12: *Comparison between the original fish model (left), the optimized version using cage deformation (center), and the optimized version using manifold harmonics (right).*

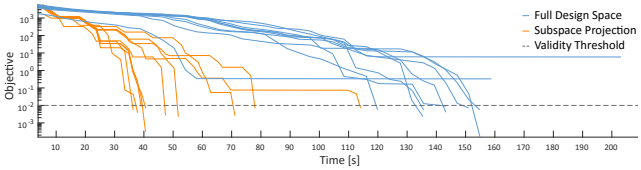


Figure 13: *Run time versus objective value for the optimization of the fish model using manifold harmonics with 10 random initial parameter values.*

the boundary shape is preserved. Fig. 11 compares the run time versus objective value graphs of cage-based optimization, with and without subspace projection. Subspace projection accelerates convergence by a factor between 2 and 3. Statistics are given in Table 3.

Our most difficult example is the optimization of a nearly radially symmetric bell model in the shape of a rabbit. Many of the eigenvalues computed as the solution to Eqs. 10 have multiplicity two in the radially symmetric case. If the symmetry is only approximate, the twin frequencies diverge slightly and produce an unfavorable frequency spectrum. Therefore one has to fix the two natural modes to the same frequency in order to optimize a single tone.

Using the subspace projection algorithm, we can optimize a total of 5 natural modes for the rabbit-shaped bell shown in Fig. 15. Due to fabrication reasons, we have used only a half-bell in order to be able to mill the shape from a solid block of aluminum. We have optimized the frequencies to the spectrum D6, F6, A6, D7, and F7. Fig. 15 shows a 3d render of the outer surface, a render of the front and back, and the milled aluminum result.

Finally, in Fig. 16 we plot of run time versus objective value for the failed attempt using a full 32-dimensional manifold harmonics design space and the converged attempt using subspace projection. A dashed line represents recomputation of the reduced space $\mathcal{R}$, and the plateau in the curve following it corresponds to the time it takes to evaluate the gradient of the property function w.r.t. all 32 parameters.

Discussion. The results presented here are inspired by the pioneering work by Bharaj et al. [2015], who were the first to computationally optimize multiple natural frequencies for the purpose of instrument fabrication. Their work focuses on accurate prediction of the frequencies using third-order finite elements and a global sampling strategy that uses local optimization in each iteration.

By contrast, the paper at hand proposes a modification of the local optimization step that, firstly, accelerates convergence compared to optimization using the full design space and, secondly, achieves convergence in a larger percentage of test cases. This was shown on a number of 3d models, different shape parameterizations, and multiple iterations using a Halton sequence [Kuipers and Niederreiter 2012] for sampling initial parameter values. By using only second-order elements, and coarser finite element meshes, we trade some accuracy in favor of very short computation times. Most of our models take between 2 and 10 minutes to converge, while a full computation cycle including frequency and amplitude optimization using the method of Bharaj et al. [2015] takes between 2-3 hours

Figure 14: *Run time of multiple iterations optimizing the fish model. The number of manifold harmonics was varied between 4 and 24 (vertical axis), and the number of optimized frequencies between 1 and 4. Non-converged iterations are marked with a $\times$. Time is given on the horizontal axis in minutes.*

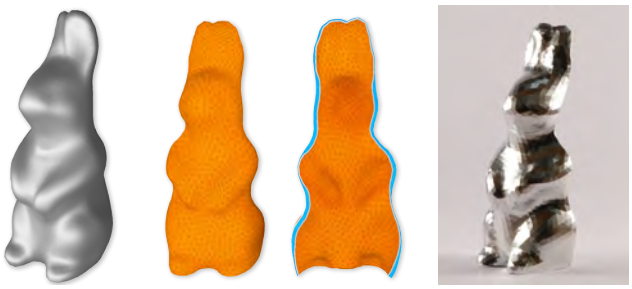


Figure 15: *Optimized rabbit bell with 5 modes. Left: render of outer surface. Center: the front an back including the mesh. Right: the result of the (half) bell milled with a 5-axis milling machine from solid aluminum.*

for a 2d example and 5-6 hours for a 3d example according to personal communication with the authors. Furthermore, our manifold harmonics parameterization avoids the strong distortions of animal silhouettes that can be observed in their results (cf. Fig. 12).

As can be seen in Fig. 9, the predicted frequencies have a near-constant negative shift with respect to the target frequencies. This suggests that the material properties for aluminum, like density and Young's modulus, used in the frequency computations deviate from the true material properties. Another source of error for our animal glockenspiel is the production error introduced by CNC milling, which lies in a range of $0.2-0.4$ mm. This process is lengthier than cutting pieces from an aluminum plate with a waterjet cutter and allowed us to do fewer production iterations to maximize accuracy.

6.2 Optimization of Mass Properties

In this section we explore the optimization of mass properties, i.e., the mass, center of gravity, and the inertia tensor [Bächer et al. 2014; Musialski et al. 2015] using subspace projection and constraint mapping on the examples of static stability and buoyancy. Control over these properties enables influencing the behavior of objects to make them stand stably, spin stably, float in water, or return to an upright position when tilted. We demonstrate that constraint mapping significantly enlarges the manifold harmonics design space compared to the version with box constraints, and that subspace projection further accelerates convergence.

Static stability. The property variables required to formulate the objectives considered here are the volume V and the center of mass $\mathbf{c} = [c_x, c_y, c_z]^T$ of a model. Given a constant material density, these quantities are defined as the volume integrals

$$V = \int_{\mathcal{V}} \mathrm{d}V \quad \text{and} \quad \mathbf{c} = \int_{\mathcal{V}} \mathbf{x}\, \mathrm{d}V,$$

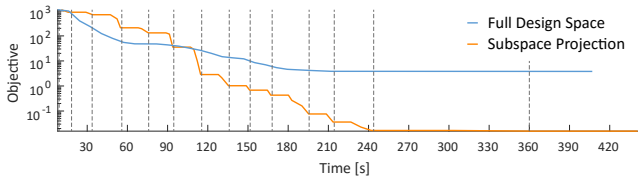


Figure 16: *Graph of run time versus objective values for the optimization of 4 natural modes of a rabbit-shaped bell.*

but can be transformed into sums of surface integrals over the triangles comprising $\overline{\mathcal{M}}$ and $\underline{\mathcal{M}}$ using the divergence theorem [Eberly 2010]. If the volume of the outer surface $V(\overline{\mathcal{M}})$ is needed, one may consider only the triangles comprising $\overline{\mathcal{M}}$. The formulation using surface integrals is also amenable to analytic differentiation with respect to a shape parameter α. s The condition for static stability, i.e., stable standing of an object, is that the center of mass projects onto the base of support along the direction of gravity $(0,0,-1)^T$. The base of support of an object is the convex hull of all its points that are touching the ground. Additional stability can be gained by placing the center of gravity as low as possible. Before optimizing a model, we place its origin at the center of the base of support. Then the objective for static stability can be written as

$$f(\boldsymbol{\alpha}) = \omega_1 \left(c_x(\boldsymbol{\alpha})^2 + c_y(\boldsymbol{\alpha})^2 \right) + \omega_2 c_z(\boldsymbol{\alpha}). \qquad (12)$$

Buoyancy. The second objective we tackle is to make an object float in a liquid in an upright position. This is made possible by equilibrating the gravitational force $-\rho \mathbf{g}$ and the buoyant force $\rho_{\text{fluid}} V(\overline{\mathcal{M}}) \mathbf{g}$ while the center of mass $\mathbf{c}$ and the center of buoyancy $\mathbf{c}_b$ are aligned in the (x,y)-plane. The center of buoyancy equals the center of mass evaluated on the outer surface $\overline{\mathcal{M}}$ of the model, without considering the triangles of $\underline{\mathcal{M}}$. An additional constraint requires that $c_{b,z}$ be higher than c_z to guarantee a stable equilibrium state.

These goals can be combined in the objective function

$$\begin{aligned} f(\boldsymbol{\alpha}) &= \sum_{i=1}^{3} \omega_i f_i, \quad \text{where} \qquad (13) \\ f_1 &= \left\| \rho_{\text{fluid}} V(\overline{\mathcal{M}}) - \rho V \right\|^2, \\ f_2 &= \| c_x - c_{b,x} \|^2 + \| c_y - c_{b,y} \|^2, \\ f_3 &= \| \max(0, c_z + t - c_{b,z}) \|^2. \end{aligned}$$

The constant t is a threshold that determines a safety margin between the heights of the center of gravity and the center of buoyancy, which we set to 2 mm.

Results and Discussion. We have implemented and compared the objectives in order to demonstrate the benefits of our method over optimization in the full design space with constraints as performed by Musialski et al. [2015]. Fig. 17 shows the results of optimizing the armadillo model for static stability and the fish model for optimized flotation in water. All models have been optimized using manifold harmonics, once without and once with subspace projection and constraint mapping.

The main insight to be gained here is that the constraint mapping approach improves the results considerably compared to the approach of Musialski et al. [2015]. One can clearly see that the solution using constraint mapping takes advantage of the increased expressiveness of the design space by introducing sharp features as part of the inner surface of the armadillo. It gives the inner void a less smooth but a more flexible shape, and resembles nearly a plane as in the results shown by Prevost et al. [2013].

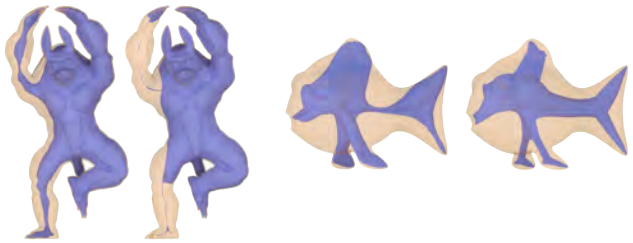


Figure 17: *Optimized armadillo and fish models: with full box-constraints (left models, armadillo not converged, fish converged), with constraints mapping and subspace projection (right models, both converged).*

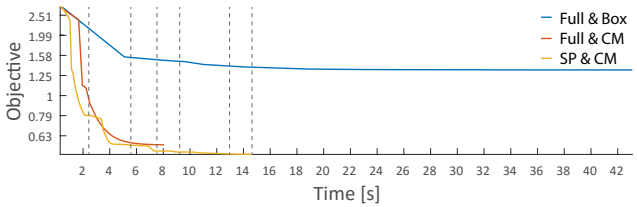


Figure 18: *Improvement of the objective value over time for the optimization of the armadillo model for static stability.*

For the optimization of the armadillo, we use a 72-dimensional manifold harmonics basis to define the design space $\mathcal{D}$. We compare the run times of three methods, as shown in Fig. 18: optimization on $\mathcal{D}$ using box constraints on the offset magnitudes (blue); optimization on $\mathcal{D}$ using constraint mapping (red); optimization using subspace projection and constraint mapping (yellow). Vertical dashed lines mark the points at which the reduced parameter space $\mathcal{R}$ was calculated anew using subspace projection. The method using box constraints (blue) does not converge to a valid solution because the design space is too restrictive. Constraint projection alleviates this problem, and the two methods using it (red and yellow) behave similarly, with subspace projection slightly outperforming optimization on the full design space.

Table 4 summarizes statistics from the optimization of the fish and the armadillo. It also includes results of a constrained formulation of the problem, in which the safety margin between the centers of gravity and buoyancy, and the projection of the center of gravity onto the base of support respectively, are modeled as non-linear hard constraints and solved via SQP.

6.3 Optimization of Structural Strength

To demonstrate the suitability of our subspace projection scheme to problems apart from frequency optimization and mass property optimization, we apply it to the optimization of structural strength. In the same vein as Lu et al. [2014], we optimize a 3d model to have minimal material cost while respecting a maximal material stress level in a given load scenario.

Objective. We measure the structural stability of a 3d model using the von Mises yield criterion [Mises 1986]. This criterion postulates that yielding of a material begins once the second invariant of the Cauchy stress tensor $\boldsymbol{\sigma}$,

$$\begin{aligned} J_2 = &\frac{1}{6} \left[(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 \right] \\ &+ \sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2, \end{aligned}$$

reaches a critical value of κ^2, where κ is the yield stress of the material in shear.

In a valid solution to the strength optimization problem, the critical stress value is approached, but not exceeded in any point throughout the object. We evaluate this condition by computing J_2 for every node in every element of the finite element mesh. This set of nonlinear constraints is transformed into the objective

$$f_J = \sum_{\substack{n \in \mathcal{N} \\ J_2(n) > J_{max}}} \| J_2(n) - J_{max} \|^2 , \qquad J_{max} = (1-\varepsilon)\kappa^2 ,$$

where $\mathcal{N}$ is the set of all nodes in all elements. The constant ε can be adjusted to penalize stresses that come too close to the critical stress value. We have found that a value of 0.05 works well. In combination with the objective to minimize material consumption, we obtain the objective function

$$f(\boldsymbol{\alpha}) = \omega_1 m(\boldsymbol{\alpha})^2 + \omega_2 f_J(\boldsymbol{\alpha}) , \tag{14}$$

where m is the mass of the object. The weights ω_1 and ω_2 are chosen to prioritize the objective f_J, such that a region of valid solutions is approached quickly by the optimization routine.

We project the design space of this problem onto a two-dimensional subspace $\mathcal{R}$, which is computed as the solution to the least-norm problem (5). The two properties $\boldsymbol{\varphi}(\boldsymbol{\alpha})$ for this problem correspond directly to the objectives m and f_J, both of which can be differentiated analytically w.r.t. the design parameters $\boldsymbol{\alpha}$. An important ingredient to the computation of these gradients is the derivative of the displacement vector $\mathbf{u}$, which is the solution to the finite-element equation $\mathbf{K}\mathbf{u} = \mathbf{F}$, with the stiffness matrix $\mathbf{K}$ and the load vector $\mathbf{F}$. Through implicit differentiation, we arrive at the solution

$$\frac{\partial \mathbf{u}}{\partial \alpha_i} = \mathbf{K}^{-1} \left(\frac{\partial \mathbf{f}}{\partial \alpha_i} - \frac{\partial \mathbf{K}}{\partial \alpha_i} \mathbf{u} \right) .$$

Based on this quantity, the gradients of J_2 and, in consequence, of f are readily determined.

Results. We perform strentgh optimization on a thin-walled 3d model of a kitten, depicted in Fig. 19, using the manifold harmonics parameterization. Table 5 collects timings of this optimization, with and without subspace projection. The external force is assumed to be a concentrated load applied to the head of the model. Originally, the model has a constant wall thickness of 1.5 mm and is predicted to yield near the point of load application.

The optimization algorithm is run on the model to double the maximum allowable load, which results in a thickening of the wall at the head and the belly of the model. Both the original and the optimized model were 3d printed from polylactic acid (PLA) and subjected to a compressive test in a universal testing machine. Fig. 19 shows the force-displacement curves of the test and photographs of the destroyed models to verify that the locations of initial yielding were predicted correctly.

Discussion. The goal of strength optimization through an automatic procedure was previously explored by Stava et al. [2012], Wang et al. [2013], and Lu et al. [2014]. The optimization algorithms proposed in these works distribute elements like struts and air bubbles throughout a model to maximize the strength-to-weight ratio. Therefore, these methods work with discrete variables, like the total number of newly introduced elements, making them ill-suited to optimization via Newton-like methods.

As noted by Stava et al. [2012], the main limiting factor of strength optimization algorithms that rely on the computation of stress gradients is the high computational expense. However, through computation of the stiffness matrix and its gradient on the GPU, and drastic reduction of the parameter set, our approach remains computationally feasible despite a large design space. It is also fully continuous and can thus be solved using a Newton-based optimization routine. However, our method is currently limited by its inability to handle topological changes.

7 Implementation and Fabrication

Implementation. All applications presented in this paper have been implemented using the linear algebra routines and optimization routines in MATLAB. To solve optimization problems, we use the quasi-Newton algorithm implemented in `fminunc` for unconstrained problems and the active-set algorithm implemented in `fmincon` for constrained problems. Sparse generalized eigenvalue problems are solved with the Lanczos algorithm implemented in `eigs` in order to find the k smallest eigenvalues with corresponding eigenvectors.

For the optimization of natural frequencies and of structural strength, we use a finite-element discretization of the governing PDEs. Since our models are described as the volume between two triangle meshes with identical topology, second-order triangular wedges are a natural choice of element. The most time-consuming operations in the optimization pipeline are the computations of the element stiffness matrices $\mathbf{K}^e$, the element mass matrices $\mathbf{M}^e$ and, most importantly, their derivatives $\partial\mathbf{K}^e/\partial\alpha_i$ and $\partial\mathbf{M}^e/\partial\alpha_i$. These computations were moved to the GPU and implemented in CUDA. Global matrix assembly is performed by the `sparse` matrix constructor in MATLAB after reading back the data from the GPU. The running time of this step was improved significantly by arranging the data in column-first order on the GPU prior to matrix construction.

Fabrication. All instruments shown in this document and in the supplemental video were manufactured on a Spinner U-620 CNC milling machine from AlCuMgPb F37, an aluminium alloy with a purity of about 95%. The milling tool paths were computed with the software SprutCAM 10. The 3d models used to test structural strength optimization and buoyancy optimization were produced from PLA on an Ultimaker 2, a 3d printer based on fused deposition modeling. The compression test on the kitten model was performed on a universal testing machine by the Zwick Roell Group (Z050).

8 Conclusions

Summary. In this paper we have proposed a novel strategy for the efficient local parameterization of underdetermined shape optimization problems. The rationale of our method is based on the observations that shape properties can be well optimized in rich design spaces, however, local preconditioning combined with dimensionality reduction improves both the speed and the convergence of local optimization. Such an approach is novel and has not been documented before to our knowledge.

Additionally, we provided a generic formulation for the optimization of various shape properties, and we demonstrated the effectiveness of our approach on three different shape optimization problems. We showed on the basis of these examples that our approach performs better than optimization in the whole design space in almost every case. Moreover, we showed that the method is independent of the chosen parameterization.

Limitations. We have applied our subspace projection method to three different problems with various models and parameterizations and showed that it performs better than optimization the full design space in almost all cases. While it converges in most cases from any initial values, it is not ensured that it always converges. It is still a local optimization approach which can be trapped in a bad local minimum.

Another limitation of our method is its dependency on a smooth and differentiable parameterization, since the subspace is derived from the gradient in the design space. We have not tested how the method behaves with non-smooth objective functions.

Finally, we have not tested if our method could be applied to topology optimization problems, like the voxel-carving approach

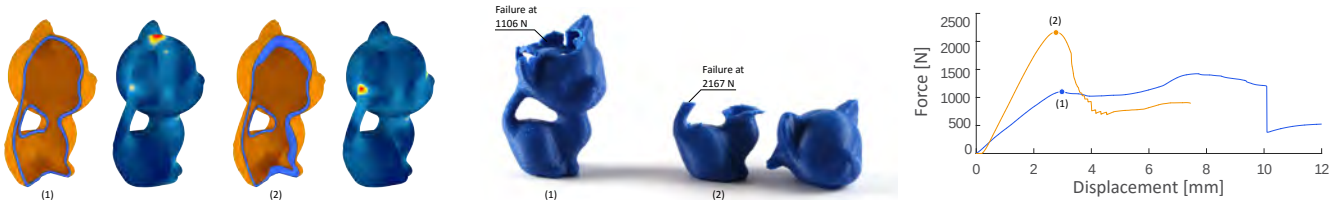


Figure 19: *(1) Original model, (2) optimized model. Left: cross-section and stress distribution under concentrated load from the top. Center: photograph of 3d printed models after compression test. Right: force-displacement diagrams with yield points marked.*

of Bächer et al. [2014]. However, it would be interesting to explore further applications domains of our method.

Future Work. For future work we plan to investigate the applicability of the subspace projection method to further optimization problems in order to broaden the range of its applications. Moreover, we plan to analyze the method in combination with other local optimization strategies, like interior point methods.

Acknowledgments

We would like to thank Thomas Koch for help with the strength tests, Thomas Auzinger for initial discussions, and the anonymous reviewers for valuable suggestions. This research was funded by the Austrian Science Fund (FWF P27972-N31), the Vienna Science and Technology Fund (WWTF, ICT15-082), the European Research Council (ERC Advanced Grant "ACROSS", grant agreement 340884), and the German Research Foundation (DFG, Gottfried-Wilhelm-Leibniz Programm).

model	algo.	variant	conv.	time	#ffe	#spfe
C7	QN (64)	Full+CM	0/5	n/a	n/a	
		SP+CM	3/5	5.5 min	5.3	27.7
	LM (64)	Full+CM	5/5	7.6 min	12.4	
		SP+CM	5/5	3.5 min	3.8	15.6
D7	QN (64)	Full+CM	1/5	7.1 min	16	
		SP+CM	5/5	2.3 min	3.4	12.2
	LM (64)	Full+CM	5/5	2.5 min	5.6	
		SP+CM	5/5	1.1 min	2	4.6
E7	QN (64)	Full+CM	1/5	6.4 min	16	
		SP+CM	5/5	3.2 min	5.2	16.4
	LM (64)	Full+CM	5/5	3.1 min	7.4	
		SP+CM	5/5	1.7 min	2.8	9.6
F7	QN (64)	Full+CM	0/5	n/a	n/a	
		SP+CM	3/5	2.9 min	4	19
	LM (64)	Full+CM	5/5	5.5 min	11.8	
		SP+CM	5/5	1.7 min	2.6	10.2
G7	QN (64)	Full+CM	2/5	14.0 min	36	
		SP+CM	2/5	3.3 min	5	25.5
	LM (64)	Full+CM	5/5	3.3 min	8.4	
		SP+CM	5/5	2.7 min	6	8.2
A7	QN (64)	Full+CM	1/5	10.1 min	18	
		SP+CM	4/5	3.8 min	4	17.5
	LM (64)	Full+CM	5/5	2.9 min	5.2	
		SP+CM	5/5	2.4 min	3.6	4.6
B7	QN (64)	Full+CM	5/5	7.0 min	16.2	
		SP+CM	4/5	2.4 min	2.8	20.8
	LM (64)	Full+CM	5/5	2.2 min	4.8	
		SP+CM	5/5	1.1 min	2	4.2
C8	QN (64)	Full+CM	2/5	17.3 min	38	
		SP+CM	2/5	7.1 min	7	61
	LM (64)	Full+CM	5/5	3.5 min	7.2	
		SP+CM	5/5	2.6 min	4.8	6.2

Table 1: *Results for glockenspiel computation with 64 shape parameters across multiple runs. QN=quasi-Newton; LM=Levenberg-Marquardt; conv.=number of converged runs; ffe=objective and gradient evaluations on full parameter space; spfe=objective and gradient evaluations on reduced parameter space. Times, #ffe, and #spfe are averaged over all converged runs.*

References

BÄCHER, M., BICKEL, B., JAMES, D. L., AND PFISTER, H. 2012. Fabricating articulated characters from skinned meshes. *ACM Transactions on Graphics 31*, 4 (jul), 1–9.

BÄCHER, M., WHITING, E., BICKEL, B., AND SORKINE-HORNUNG, O. 2014. Spin-It: Optimizing Moment of Inertia for Spinnable Objects. *ACM Transactions on Graphics 33*, 4 (jul), 1–10.

BATHE, K.-J. 2006. *Finite Element Procedures*. Prentice Hall.

BHARAJ, G., LEVIN, D. I. W., TOMPKIN, J., FEI, Y., PFISTER, H., MATUSIK, W., AND ZHENG, C. 2015. Computational design of metallophone contact sounds. *ACM Transactions on Graphics 34*, 6 (oct), 1–13.

BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics 29*, 4 (jul), 1.

BOTSCH, M., AND KOBBELT, L. 2005. Real-Time Shape Editing using Radial Basis Functions. *Comput. Graph. Forum 24*, 3, 611–621.

DELFOUR, M. C., AND ZOLÉSIO, J. P. 2011. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, Second Edition*. Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

EBERLY, D. H. 2010. *Game physics*, 2. edition ed. Morgan Kaufmann, Burlington, Mass.

FINSTERLE, S., AND KOWALSKY, M. B. 2011. A truncated Levenberg–Marquardt algorithm for the calibration of highly parameterized nonlinear models. *Computers & Geosciences 37*, 6 (jun), 731–738.

HAFNER, C., MUSIALSKI, P., AUZINGER, T., WIMMER, M., AND KOBBELT, L. 2015. Optimization of natural frequencies for fabrication-aware shape modeling. In *ACM SIGGRAPH 2015 Posters on - SIGGRAPH '15*, ACM Press, New York, New York, USA, 1–1.

IPSEN, I. C. F., KELLEY, C. T., AND POPE, S. R. 2011. Rank-Deficient Nonlinear Least Squares Problems and Subset Selection. *SIAM Journal on Numerical Analysis 49*, 3 (jan), 1244–1266.

KUIPERS, L., AND NIEDERREITER, H. 2012. *Uniform Distribution of Sequences*. Dover Books on Mathematics. Dover Publications.

LU, L., CHEN, B., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., AND COHEN-OR, D. 2014.

model	algo.	variant	conv.	time	#ffe	#spfe
Fish	QN (18)	Full+CM	8/10	2.4 min	36.8	
		SP+CM	10/10	0.9 min	4.3	19.3

Table 2: *Results of fish-bar optimization using 10 different initial values and 18 shape parameters. For abbreviations see Table 1.*

model	algo.	variant	conv.	time	#ffe	#spfe
Fish	QN (15)	Full	1/1	36.8 s	12	
		SP	1/1	20.4 s	2	10
Giraffe	QN (15)	Full	1/1	85.2 s	14	
		SP	1/1	44.9 s	2	12
Elephant	QN (15)	Full	1/1	202.4 s	29	
		SP	1/1	49.6 s	2	17

Table 3: *Results for 9-node cage-based deformation on different models. For abbreviations see Table 1.*

Build-to-Last: Strength to Weight 3D Printed Objects. *ACM Transactions on Graphics 33*, 4 (jul), 1–10.

MISES, R. V. 1986. The mechanics of solids in the plastically-deformable state.

MUSIALSKI, P., AUZINGER, T., BIRSAK, M., WIMMER, M., AND KOBBELT, L. 2015. Reduced-Order Shape Optimization Using Offset Surfaces. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2015) 34*, 4 (jul), 102:1–102:9.

NOCEDAL, J., AND WRIGHT, S. 2006. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York.

PANETTA, J., ZHOU, Q., MALOMO, L., PIETRONI, N., CIGNONI, P., AND ZORIN, D. 2015. Elastic textures for additive fabrication. *ACM Transactions on Graphics 34*, 4 (jul), 135:1–135:12.

PÉREZ, J., THOMASZEWSKI, B., COROS, S., BICKEL, B., CANABAL, J. A., SUMNER, R., AND OTADUY, M. A. 2015. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics 34*, 4 (jul), 138:1–138:12.

PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Transactions on Graphics 32*, 4 (jul), 1.

SCHUMACHER, C., BICKEL, B., RYS, J., MARSCHNER, S., DARAIO, C., AND GROSS, M. 2015. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics 34*, 4 (jul), 136:1–136:13.

SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics 32*, 4 (jul), 1.

SKOURAS, M., THOMASZEWSKI, B., KAUFMANN, P., GARG, A., BICKEL, B., GRINSPUN, E., AND GROSS, M. 2014. Designing inflatable structures. *ACM Transactions on Graphics 33*, 4 (jul), 1–10.

STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3D printable objects. *ACM Transactions on Graphics 31*, 4 (jul), 1–11.

TAGLIASACCHI, A., ALHASHIM, I., OLSON, M., AND ZHANG, H. 2012. Mean Curvature Skeletons. *Computer Graphics Forum 31*, 5 (aug), 1735–1744.

UMETANI, N., MITANI, J., AND IGARASHI, T. 2010. Designing Custom-made Metallophone with Concurrent Eigenanalysis. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*.

model	algo.	variant	conv.	time	#ffe	#spfe
Fish	QN (72)	Full+Box	1/1	31.9 s	47	
	QN (72)	Full+CM	1/1	2.81 s	33	
	QN (72)	SP+CM	1/1	1.77 s	2	34
	SQP (72)	SP+CM+NLC	1/1	2.12 s	3	37
Armadillo	QN (72)	Full+Box	0/1	42.5 s	67	
	QN (72)	Full+CM	1/1	11.80 s	43	
	QN (72)	SP+CM	1/1	10.96 s	6	72
	SQP (72)	SP+CM+NLC	1/1	5.26 s	3	33

Table 4: *Results for optimization of mass properties with 72 parameters. The fish is optimized for buoyancy, and the armadillo for static stability. NLC refers to non-linear hard constraints that require the correct positioning of the center of buoyancy, and the projection of the center of gravity inside the support polygon respectively. For abbreviations see Table 1.*

model	algo.	variant	conv.	time	#ffe	#spfe
Kitten	QN (24) / u.v.	Full	1/1	10.7 min	21	
		SP+CM	1/1	4.8 min	4	27
	QN (24) / u.o.	Full	1/1	40.0 min	79	
		SP+CM	1/1	8.0 min	8	39

Table 5: *Results for the optimization of structural strength. The "u.v." ("until valid") rows summarize performance until the solver finds the first solution meeting the stress requirements. For the "u.o." ("until optimal") rows, the solver is run until a local optimum for the minimization of mass is found. For abbreviations see Table 1.*

UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Transactions on Graphics 30*, 4 (jul), 1.

UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics 31*, 4 (jul), 1–11.

VALLET, B., AND LÉVY, B. 2008. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum 27*, 2 (apr), 251–260.

WANG, L., AND WHITING, E. 2016. Buoyancy Optimization for Computational Fabrication. *Computer Graphics Forum (Proceedings of Eurographics 2016) 35*, 2.

WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3D objects with skin-frame structures. *ACM Transactions on Graphics 32*, 6 (nov), 1–10.

ZHOU, Q., PANETTA, J., AND ZORIN, D. 2013. Worst-case structural analysis. *ACM Transactions on Graphics 32*, 4 (jul), 1.

ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Transactions on Graphics 31*, 6 (nov), 1.