

Spin-It Faster: Quadrics Solve All Topology Optimization Problems That Depend Only On Mass Moments

CHRISTIAN HAFNER, ISTA, Austria

MICKAËL LY, ISTA, Austria

CHRIS WOJTAN, ISTA, Austria

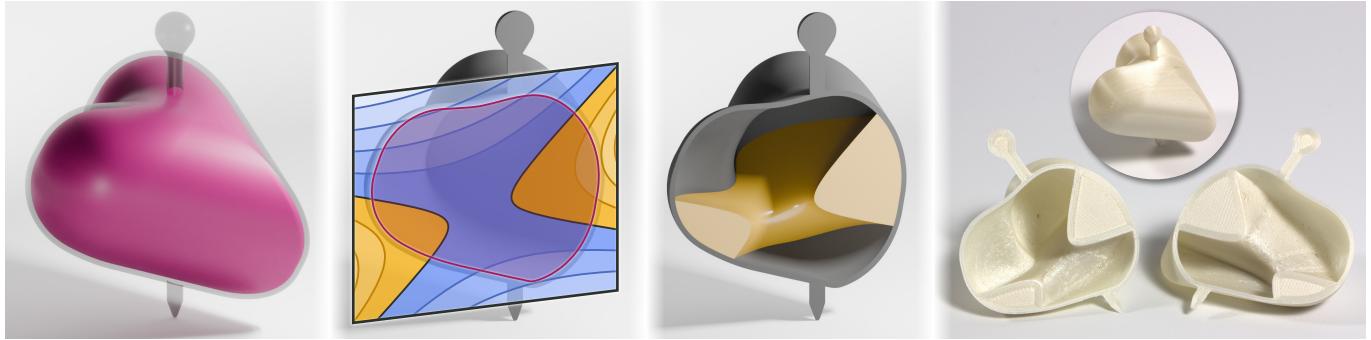


Fig. 1. **Overview.** *Left:* Spinnability of a spinning top can be ensured by solving a topology optimization problem on the interior (pink) of a target shape (transparent-gray). *Center-left:* We show that the optimal mass distribution is always obtained by placing material where a degree-two polynomial attains positive values (yellow), and leaving the rest of the domain empty (blue). Isolevels of the polynomial are shown on a cutoff plane. *Center-right:* This results in a material-air interface shaped like a quadric; in this example, a hyperboloid (yellow). *Right:* A physical prototype, 3d printed in two parts and then glued.

The behavior of a rigid body primarily depends on its mass moments, which consist of the mass, center of mass, and moments of inertia. It is possible to manipulate these quantities without altering the geometric appearance of an object by introducing cavities in its interior. Algorithms that find cavities of suitable shapes and sizes have enabled the computational design of spinning tops, yo-yos, wheels, buoys, and statically balanced objects. Previous work is based, for example, on topology optimization on voxel grids, which introduces a large number of optimization variables and box constraints, or offset surface computation, which cannot guarantee that solutions to a feasible problem will always be found.

In this work, we provide a mathematical analysis of constrained topology optimization problems that depend only on mass moments. This class of problems covers, among others, all applications mentioned above. Our main result is to show that no matter the outer shape of the rigid body to be optimized or the optimization objective and constraints considered, the optimal solution always features a quadric-shaped interface between material and cavities. This proves that optimal interfaces are always ellipsoids, hyperboloids, paraboloids, or one of a few degenerate cases, such as planes.

This insight lets us replace a difficult topology optimization problem with a provably equivalent non-linear equation system in a small number (<10) of variables, which represent the coefficients of the quadric. This system can be solved in a few seconds for most examples, provides insights into the

Authors' addresses: Christian Hafner, ISTA, Am Campus 1, Klosterneuburg, 3400, Austria, chafner@ista.ac.at; Mickaël Ly, ISTA, Am Campus 1, Klosterneuburg, 3400, Austria, mickael.ly@ista.ac.at; Chris Wojtan, ISTA, Am Campus 1, Klosterneuburg, 3400, Austria, wojtan@ist.ac.at.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

0730-0301/2024/7-ART78

<https://doi.org/10.1145/3658194>

geometric structure of many specific applications, and lets us describe their solution properties. Finally, our method integrates seamlessly into modern fabrication workflows because our solutions are analytical surfaces that are native to the CAD domain.

CCS Concepts: • Mathematics of computing → Topology; • Computing methodologies → Shape modeling; • Theory of computation → Computational geometry.

Additional Key Words and Phrases: Topology Optimization, Mass Moments, Computational Geometry

ACM Reference Format:

Christian Hafner, Mickaël Ly, and Chris Wojtan. 2024. Spin-It Faster: Quadrics Solve All Topology Optimization Problems That Depend Only On Mass Moments. *ACM Trans. Graph.* 43, 4, Article 78 (July 2024), 13 pages. <https://doi.org/10.1145/3658194>

1 INTRODUCTION

The mass, center of mass, and inertia tensor of a three-dimensional body completely determine its rigid-body behavior, as dictated by Newton's second law of motion. These *mass moments* are comprised of ten scalars, and encompass all information needed to evaluate properties such as the static stability of an object in a given orientation, or the stability of a rotational motion around a given axis. Hence, controlling the values of mass moments is critical for the design of floor stands, suspended fixtures, wheels, buoys, and a variety of toys, such as yo-yos, spinning tops, and roly-poly dolls.

Advances in digital fabrication methods such as 3d-printing have simplified the manufacture of objects with a high geometric and topological complexity. In particular, this facilitates the introduction of cavities into the interior of an object, which can be used to control its mass moments without altering the outward appearance.

Motivated by this prospect, Bächer et al. [2014] posed the following computational design question: Can we find a binary mass distribution inside a given three-dimensional body, such that the resulting object will solve a constrained topology optimization problem that depends only on mass moments?

This task can be formalized as follows: Given a design domain $\Omega \subset \mathbb{R}^3$, determine among all subsets $\omega \subset \Omega$ the one that minimizes

$$f(m_\omega, c_\omega, I_\omega) \quad \text{s.t.} \quad 0 = g_i(m_\omega, c_\omega, I_\omega) \text{ for all } i = 1, \dots, k, \quad (1)$$

where m_ω , c_ω , and I_ω denote the mass, center of mass, and inertia tensor of ω , respectively. The functions f and g_1, \dots, g_k describe the optimization objective and constraints, and depend on ω only via these mass moments. The solution ω can be interpreted as a binary mass distribution on Ω by considering ω to be filled by a medium of constant density, and $\Omega \setminus \omega$ to be empty.

Our main contribution is to show a universality property for problems of this type: *The interface of the optimal mass distribution is always the solution of a degree-two polynomial equation on Ω* , regardless of the domain, objective, and constraints considered. The interfaces that arise this way are shaped like *quadrics*, a family of surfaces comprised of ellipsoids, hyperboloids, and paraboloids, as well as degenerate quadrics, which include double cones and planes among others. This characterization of solutions holds under very general assumptions that amount only to the differentiability of f and g_i , and a non-degeneracy property of the optimization problem.

Besides providing a geometric insight, our result is immediately useful for finding numerical solutions: It allows us to reduce the search space to a small parametrized family of surfaces, namely that of all quadrics, while preserving the optimal solution to the continuous topology optimization problem. By contraposition, this can also give us a certificate of infeasibility for the original problem: If there is no feasible solution among the quadrics, there will be no feasible solution at all.

Our method also fits neatly into a computational design and fabrication pipeline, because the result is an analytical surface that can be processed natively in CAD software. If the three-dimensional object was originally designed using CAD, this avoids switching representations after optimization and can significantly simplify post-processing steps such as partitioning the object to prepare a 3d-print or modeling an assembly.

1.1 Related Work

Fabrication-aware design research has discovered many different uses for the ability of 3d printers to produce objects that are partially hollow. Wang et al. [2013] and Lu et al. [2014] add large cavities in order to improve the strength-to-weight ratio of 3d models and to make 3d printing more cost-effective. Wu et al. [2017] take this idea further by computing a fine network of cavities that mimic the porous structure of bones. Another line work uses spatially-varying microstructures to achieve controlled elastic deformations of 3d-printed objects [Ion et al. 2016; Panetta et al. 2015]. More applications that focus on intricate cavity systems can be found in a survey [Feng et al. 2018].

Optimization of mass moments is a particular goal that can be achieved through the introduction of cavities. In graphics, many works have explored this topic to optimize specific objectives, and to

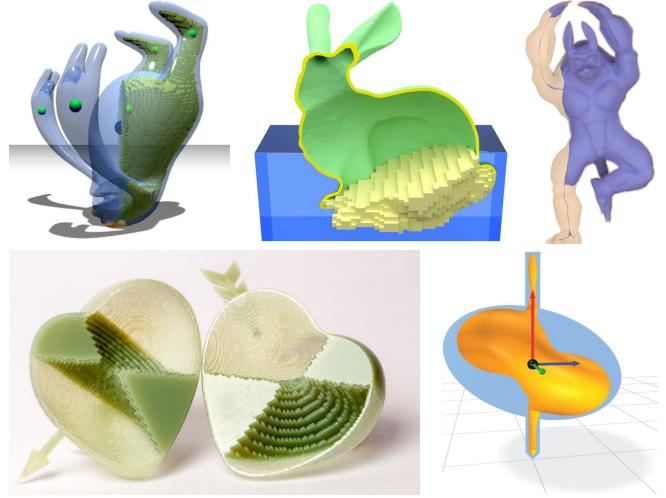


Fig. 2. **Previous Work.** Top: Results for static and buoyant stability from Prévost et al. [2013], Wang and Whiting [2016], and Musalski et al. [2016]. Bottom: Spinning tops from Bächer et al. [2014] and Musalski et al. [2015].

design 3d objects with specific functionalities. A popular approach is to represent the interior of a 3d model using a voxel grid, and employ density-based **topology optimization** [Bendsoe and Sigmund 2003; Deaton and Grandhi 2014] to decide in which voxels to deposit material. Prévost et al. [2013] optimize the center of mass to enable a 3d-printed object to stand stably on a small contact area, or to hang in a prescribed orientation from a string. Bächer et al. [2014] optimize a non-linear objective that depends on the center of mass and moments of inertia in order to produce yo-yos and spinning tops with asymmetric shapes. Furthermore, they solve the topology optimization problem hierarchically using an octree in order to accelerate the process. Wang and Whiting [2016] enable 3d-printed objects to float at a particular height and orientation when partially submerged in water by optimizing for stable hydrostatic equilibria. Zhao et al. [2016] propose a method for designing so-called roly-poly dolls by hollowing a 3d model in order to control its center of mass, and attaching it to a rounded base.

The topology-optimization approach is very general in that it can solve for cavities with arbitrary shape and topology, but it leads to a large-scale optimization problem with $>10,000$ variables, even for a moderate grid resolution of $25 \times 25 \times 25$. Furthermore, the variables are subject to box constraints, and the objective function is often non-linear, which necessitates sophisticated optimization algorithms to achieve high performance.

Another approach to mass-moment optimization is to define the shape of a cavity as an **offset surface** of the model boundary towards the medial axis [Musalski et al. 2015, 2016]. Using a reduced basis, this leads to an optimization problem in ~ 100 variables. The downside of this approach is that the topology and approximate shape of the cavity are fixed a priori, and that offsets cannot go beyond the medial axis. As a consequence, it is not guaranteed that a feasible solution exists in this formulation, even if the topology optimization problem was originally feasible.

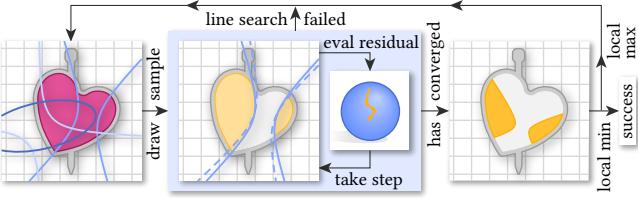


Fig. 3. **Numerical Algorithm.** *Left:* We draw a random quadric that satisfies the minimizing properties derived for a particular application, see Section 4. *Center:* We optimize the quadric parameters iteratively by computing the residual of Eq. 14 and taking a Newton step on the $(n-1)$ -sphere, see Section 5. *Right:* The algorithm terminates successfully if a local minimum was found; otherwise, it restarts from a new random sample.

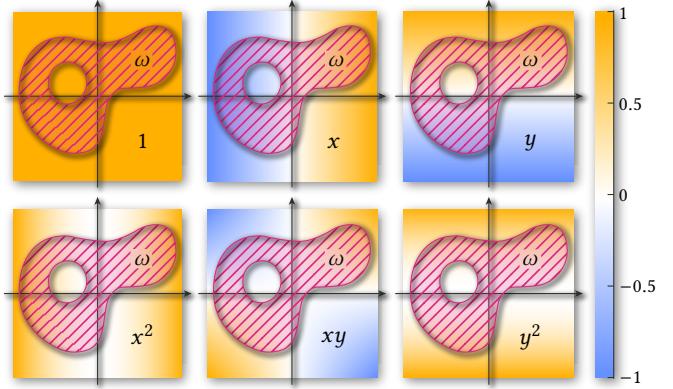


Fig. 4. **Raw Moments.** The raw moments are computed by integrating constant and linear (*top*) as well as quadratic and bilinear (*bottom*) polynomials over the domain ω (pink) with characteristic function χ , here for $d = 2$.

Some results from previous work are shown in Fig. 2. It is striking how closely the voxel-based solutions approximate either a planar interface (*top-left & top-center*), or a hyperboloid of one sheet (*bottom-left*). This is less evident for the solutions based on offset surfaces, but a resemblance to a planar interface (*top-right*) and a strongly smoothed-out hyperboloid (*bottom-right*) can be imagined.

In this work, we show that the true optimizers of all mass-moment problems previously studied are indeed quadrics, so this resemblance is not a coincidence. We leverage this insight to drastically reduce the effort needed to solve problems of this type by directly determining the coefficients of the optimal quadric from a small system of equations. This way, we can avoid solving a large-scale optimization problem and are still guaranteed to recover the true solution up to numerical tolerances.

2 OVERVIEW

Our core contribution is a mathematical analysis of topology optimization problems whose objective function and constraints depend only on the mass moments of the rigid body that is to be optimized. The main insight is that the optimal mass distribution is always such that the interface between material and void forms a quadric, which we show in Section 3. We also derive a set of optimality conditions for the coefficients of the quadric, see Eq. 7, which is the basis for our numerical solution procedure.

In Section 4, we apply our theory to problems studied in previous work, ranging from static and buoyant stability to fidget spinners, yo-yos, and spinning tops. This results in concrete optimality conditions for each application, and leads to geometric insights such as the quadric type of the solution.

Section 5 discusses our numerical method, which is a variant of Newton’s root-finding algorithm to solve the system of optimality conditions. It also discusses a practical method for evaluating the mass moments of a rigid body whose geometry arises by intersecting a triangle mesh with a quadric. The steps of the numerical solution procedure are shown schematically in Fig. 3. Finally, Section 6 shows results of applying our numerical method to two- and three-dimensional geometry.

3 OPTIMALITY CONDITIONS

In this section, we show that the topology optimization problem formulated in the introduction is always solved by a binary mass distribution whose interface is a quadric. More generally, for problems in dimension d , the interface is always the solution of a degree-two polynomial equation on the domain $\Omega \subset \mathbb{R}^d$. This insight can be used to transform the original problem into a system of non-linear equations in a small number of unknowns.

We first rephrase the topology optimization problem in terms of notation that simplifies the exposition. Then we give necessary and sufficient versions of our optimality conditions, which differ by a small gap that accounts for solutions where constraint qualifications are not met. The proof of the sufficient conditions uses only elementary methods, so we give it in full. Proving the necessary conditions is more technical, so we relegate part of the proof to the supplemental document and describe the important steps here.

3.1 Notation

We restate the topology optimization problem from Eq. 1 in a slightly modified form, and generalize the design domain to $\Omega \subset \mathbb{R}^d$. The choice of dimension $d \in \{1, 2, 3\}$ allows for the design of linear, flat, and volumetric objects. Instead of considering the subset $\omega \subset \Omega$ as the unknown directly, we encode it as a characteristic function $\chi : \Omega \rightarrow \{0, 1\}$, such that $\chi(x) = 1$ if $x \in \omega$, and $\chi(x) = 0$ otherwise. Using χ , the constant, linear, and quadratic *raw moments* of ω can be computed as¹

$$v(\chi) = \int_{\Omega} \chi, \quad \ell(\chi) = \int_{\Omega} \chi \cdot \text{id}, \quad \text{and} \quad Q(\chi) = \int_{\Omega} \chi \cdot \text{id} \otimes \text{id}, \quad (2)$$

where id denotes the identity map on \mathbb{R}^d . For $d = 3$, and written in coordinates, these definitions are equivalent to

$$v = \int_{\omega} 1 \, dV, \quad \ell = \int_{\omega} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \, dV, \quad \text{and} \quad Q = \int_{\omega} \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} \, dV,$$

¹Our notation omits the variable of integration when it is not needed. For example, the expression for the linear moment expands to $\ell(\chi) = \int_{\Omega} \chi \cdot \text{id} = \int_{\Omega} \chi(\vec{x}) \cdot \text{id}(\vec{x}) \, d\vec{x} = \int_{\Omega} \chi(\vec{x}) \cdot \vec{x} \, d\vec{x} = \int_{\omega} \vec{x} \, d\vec{x}$.

where $dV = dV(x, y, z)$. Fig. 4 illustrates the individual components of this computation for $d = 2$. Note that the definitions imply $v(\chi) \geq 0$, $\ell(\chi) \in \mathbb{R}^d$, and $Q(\chi) \in \mathbb{R}^{d \times d}$ symmetric positive-semi definite.

We collect the independent entries of v , ℓ , and Q in the vector $r \in \mathbb{R}^n$ of raw moments. This gives a total of $n = 1 + d + \frac{d(d+1)}{2}$ components because we omit the redundant entries of the symmetric matrix Q . For $d = 3$, we have $n = 1 + 3 + 6 = 10$, and

$$r = (v, \ell_1, \ell_2, \ell_3, Q_{11}, Q_{22}, Q_{33}, Q_{23}, Q_{13}, Q_{12})^\top. \quad (3)$$

As discussed further in Section 4.1, the mass m_ω , center of mass c_ω , and inertia tensor I_ω , which were used in the introduction, can be expressed in terms of r . This allows us to rewrite the optimization objective and constraints to depend on r directly, instead of on m_ω , c_ω , and I_ω . This yields the *topology optimization problem*

$$\min_{\chi: \Omega \rightarrow \{0,1\}} f(r(\chi)) \quad \text{s.t.} \quad 0 = g_i(r(\chi)) \text{ for all } i = 1, \dots, k, \quad (\text{TOP})$$

with $f, g_1, \dots, g_k : \mathbb{R}^n \rightarrow \mathbb{R}$ assumed differentiable. The unknown in this problem is the subset $\omega \subset \Omega$ encoded by its characteristic function χ such that $\omega = \chi^{-1}(1)$. This subset will be interpreted as the portion of Ω filled by a solid material, and $\Omega \setminus \omega$ as a void.

Specific optimization problems, such as optimizing stably standing objects or spinning tops, can be obtained from this general form by choosing a specific objective function f and constraint functions g_1, \dots, g_k that depend only on the moments of ω . We will study these specific applications in Section 4. The remainder of this section tackles the general problem (TOP), so any property that we show will automatically apply to all these specific applications.

3.2 Optimality Conditions

Our goal is to show that for every instance of (TOP), no matter the specific choice of f, g_1, \dots, g_k , the optimal solution can be characterized as follows: The optimal subset $\omega \subset \Omega$ consists of exactly the points where some degree-two polynomial in d variables is positive. In two dimensions, the shapes that arise this way are exactly the *conic sections*: ellipses (see Fig. 5), parabolas, and hyperbolas; and a few degenerate cases such as lines and line pairs. In three dimensions, the optimal solutions are exactly the *quadratics*: ellipsoids, paraboloids, and hyperboloids; and degenerate cases such as double cones and planes.

This result simplifies the process of solving (TOP) dramatically: It allows us to exchange the infinite-dimensional search space of functions $\chi : \Omega \rightarrow \{0, 1\}$ with a low-dimensional set of shapes that is easy to parametrize and optimize over. Below we introduce our parametrization and present a local optimality condition that identifies the specific parameter values that optimally solve an instance of (TOP), and which is used in our numerical method.

To parametrize $\omega \subset \Omega$, we introduce a non-vanishing polynomial $\beta : \Omega \rightarrow \mathbb{R}$ of degree at most two, which can be written as

$$\beta(x) = a + \langle b, x \rangle + \langle x, Ax \rangle, \quad (4)$$

with $a \in \mathbb{R}$, $b \in \mathbb{R}^d$, and $A \in \mathbb{R}^{d \times d}$ a symmetric matrix. We collect these coefficients in a vector $p \in \mathbb{R}^n \setminus \{0\}$ of unknowns, such that the ordering matches that from Eq. 3. For $d = 3$,

$$p = (a, b_1, b_2, b_3, A_{11}, A_{22}, A_{33}, A_{23}, A_{13}, A_{12})^\top. \quad (5)$$

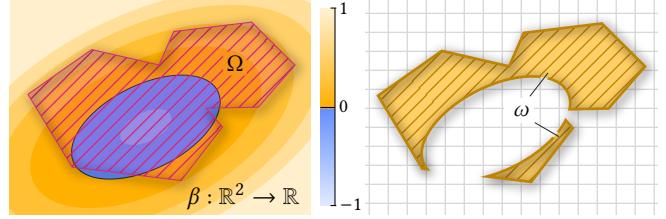


Fig. 5. Mass Distribution. Given a design domain $\Omega \subset \mathbb{R}^2$ (pink) and a degree-two polynomial β whose level sets are shown (left), a new domain $\omega \subset \Omega$ (yellow) is constructed by keeping only the points $x \in \Omega$ where $\beta(x) > 0$ (right). This domain has characteristic function χ , see Eq. 6.

The characteristic function χ of ω is given by

$$\chi(x) := \begin{cases} 1 & \text{if } \beta(x) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

which is equivalent to $\omega = \{x \in \Omega : \beta(x) > 0\}$. Note that any positive multiple of p will generate the same distribution, so we may choose $p \in S^{n-1} \subset \mathbb{R}^n$, that is, $\|p\| = 1$.

Assume that χ is generated from p through this procedure. We then show that χ is locally optimal for (TOP) if

$$\exists \mu > 0, \lambda_1, \dots, \lambda_k \in \mathbb{R} : \nabla_r f = -\mu p + \sum_{i=1}^k \lambda_i \nabla_r g_i. \quad (7)$$

This is a non-linear system of n equations which is reminiscent of standard first-order optimality conditions. However, instead of containing derivatives with respect to the vector p of unknowns, the gradients $\nabla_r f, \nabla_r g_i \in \mathbb{R}^n$ are taken with respect to the vector r of raw moments, and evaluated at $r(\chi)$. This has a significant advantage for computation: By solving Eq. 7 with Newton's method, we need only evaluate first-order derivatives with respect to p but achieve *quadratic convergence* nonetheless, as described in Section 5.

The remainder of this section is concerned with presenting a proof of the necessity and sufficiency of Eq. 7 for solving (TOP). Readers primarily interested in applications and an implementation of the numerical method may skip ahead to Section 4.

3.3 Sufficiency

Our main tool for studying (TOP) is the *relaxed problem*

$$\min_{\chi: \Omega \rightarrow [0,1]} f(r(\chi)) \quad \text{s.t.} \quad 0 = g_i(r(\chi)) \text{ for all } i = 1, \dots, k, \quad (\text{RP})$$

which is identical to (TOP) except that the codomain of χ is $[0, 1]$ instead of $\{0, 1\}$. This is a true relaxation in the sense that any χ which is feasible for (TOP) is also feasible for (RP), and has the same objective value. In consequence, any local optimizer χ^* of (RP) which is feasible for (TOP) is also locally optimal for (TOP).

Our optimality conditions turn out to be applicable even to (RP), which is more amenable to analysis than (TOP). Thus, we show both the necessary and sufficient version of the conditions for (RP), and let them carry over to (TOP). The following result shows that Eq. 7 is sufficient for local minima of (RP), and distinguishes them from local maxima and saddle points despite being only first-order:

THEOREM 1. Let $p \in S^{n-1}$ and $\chi^* : \Omega \rightarrow \{0, 1\}$ defined by Eqs. 4–6, and assume that $0 = g_i(r(\chi^*))$ for all $i = 1, \dots, k$ and Eq. 7 is satisfied. Then χ^* is a strict local optimizer for (RP).

PROOF. It suffices to show that all non-vanishing, admissible variations of χ^* strictly increase the objective value. A one-parametric smooth family of such variations takes the form $\chi_\varepsilon : \Omega \rightarrow [0, 1]$ with parameter $\varepsilon \in [0, h)$ for some $h > 0$, such that $\chi_0 = \chi^*$, and $0 = g_i(r(\chi_\varepsilon))$ for all $i = 1, \dots, k$ and $\varepsilon \in [0, h)$. The assumption $0 \leq \chi_\varepsilon \leq 1$ for all $\varepsilon \in [0, h)$ implies that the first variation $\delta\chi(x) = \frac{\partial}{\partial \varepsilon} \chi_\varepsilon(x)|_{\varepsilon=0}$ satisfies, for all $x \in \Omega$,

$$\delta\chi(x) \begin{cases} \leq 0 & \text{if } \chi^*(x) = 1, \\ \geq 0 & \text{if } \chi^*(x) = 0, \end{cases} \text{ and so, } \delta\chi(x) \begin{cases} \leq 0 & \text{if } \beta(x) > 0, \\ \geq 0 & \text{otherwise,} \end{cases} \quad (8)$$

by Eq. 6. The entries of r as given by Eq. 2 are linear in χ , so the entries of the first variation δr are given by

$$\delta v = \int_{\Omega} \delta\chi, \quad \delta\ell = \int_{\Omega} \delta\chi \cdot \text{id}, \quad \text{and} \quad \delta Q = \int_{\Omega} \delta\chi \cdot \text{id} \otimes \text{id}. \quad (9)$$

Using the chain rule, we can write the first variations of f and g_i as

$$\delta f = \langle \nabla_r f, \delta r \rangle, \quad \text{and} \quad \delta g_i = \langle \nabla_r g_i, \delta r \rangle, \quad (10)$$

and we know that $\delta g_i = 0$ for all $i = 1, \dots, k$ because the family of variations obeys the equality constraints.

The last step is to show that $\delta\chi$ strictly increases the objective value. To show this, compute

$$\begin{aligned} \delta f &= \delta f - \sum_{i=1}^k \lambda_i \delta g_i \stackrel{(10)}{=} \langle \nabla_r f - \sum_{i=1}^k \lambda_i \nabla_r g_i, \delta r \rangle \stackrel{(7)}{=} -\mu \langle p, \delta r \rangle \\ &\stackrel{(5,9)}{=} -\mu \int_{\Omega} \delta\chi \cdot (a + \langle b, \text{id} \rangle + A : \text{id} \otimes \text{id}) \stackrel{(4)}{=} -\mu \int_{\Omega} \delta\chi \cdot \beta, \end{aligned}$$

where “:” denotes double contraction between matrices, and the superscripts reference the equations used for every step.

To verify $\delta f > 0$, we analyze the last expression: According to Eqs. 7 and 8, $-\mu$ is negative, and the integrand $\delta\chi \cdot \beta$ is non-positive everywhere. The latter is even strictly negative on a set of positive measure because $\delta\chi$ is non-vanishing on a set of positive measure by assumption, and β vanishes only on a measure-zero set because it is a polynomial with coefficients $p \neq 0$. Thus, the integral $\int_{\Omega} \delta\chi \cdot \beta$ has a strictly negative value, which concludes the proof. \square

Note that this result is stronger than what is usually achieved with optimality conditions involving only first-order derivatives: By satisfying Eq. 7, we conclude a strict increase of f under non-vanishing variations, rather than stationarity of χ^* . The critical assumption to achieve this is $p \neq 0$, and thus β is not the zero function: When we develop our numerical method, we will explicitly compute $p \in S^{n-1}$, so we can verify this for individual solutions.

3.4 Necessity

For the necessary variant of the optimality conditions, we have to distinguish between two solution types corresponding to $p = 0$ and $p \neq 0$. The case $p \neq 0$ leads to the same isolated strict local optima that were described by Theorem 1, and binarity of the solution emerges as a natural consequence even though it is not imposed a priori. The case $p = 0$ accounts for solutions to instances of (RP)

that are degenerate in the sense that the box constraint $\chi(x) \in [0, 1]$ is *non-binding* for all $x \in \Omega$. This means that the same solution is obtained even if (RP) is modified to allow any $\chi : \Omega \rightarrow \mathbb{R}$.

A rigorous proof of the case $p \neq 0$ is fairly technical and relies on a version of the Karush–Kuhn–Tucker (KKT) theorem for optimization problems in infinite-dimensional Banach spaces. Below, we choose the KKT optimality conditions arising from this theorem as a starting point and show the remainder of the proof, which elucidates the most important aspects. The derivation of the KKT conditions themselves can be found in the supplemental document.

THEOREM 2. Let $\chi^* : \Omega \rightarrow [0, 1]$ a local optimizer for (RP). Then one of the following holds:

- (1) There exists $p \in S^{n-1}$ such that $\chi : \Omega \rightarrow \{0, 1\}$ as defined by Eqs. 4–6 agrees with χ^* except on a measure-zero set, and Eq. 7 holds. Furthermore, χ^* is a strict local optimizer.
- (2) The solution χ^* is first-order optimal for (RP) even after modifying it to allow any $\chi : \Omega \rightarrow \mathbb{R}$.

PROOF. The necessary KKT optimality conditions applied to (RP) are as follows: There exist Lagrange multipliers $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ corresponding to the equality constraints, and functional Lagrange multipliers $\beta_0 : \Omega \rightarrow \mathbb{R}_{\geq 0}$ and $\beta_1 : \Omega \rightarrow \mathbb{R}_{\leq 0}$ corresponding to the pointwise inequality constraints $\chi(x) \geq 0$ and $\chi(x) \leq 1$ for all $x \in \Omega$ such that

$$\partial_v \mathcal{L} + \langle \partial_{\ell} \mathcal{L}, \text{id} \rangle + \partial_Q \mathcal{L} : \text{id} \otimes \text{id} = \beta_0 + \beta_1, \quad (11)$$

with $\mathcal{L} := f - \sum_{i=1}^k \lambda_i g_i$. Furthermore, $\beta_0(x)$ and $\beta_1(x)$ may only be non-zero if $\chi^*(x) = 0$ and $\chi^*(x) = 1$, respectively. In Eq. 11, $\partial_{\ell} \mathcal{L}$ and $\partial_Q \mathcal{L}$ denote the vector and matrix of partial derivatives of \mathcal{L} with respect to the entries of ℓ and Q .

We define $\beta := -\beta_0 - \beta_1$ and see that, for all $x \in \Omega$,

$$\beta(x) \begin{cases} \leq 0 & \text{if } \chi^*(x) = 0, \\ \geq 0 & \text{if } \chi^*(x) = 1, \end{cases} \text{ and so, } \chi^*(x) = \begin{cases} 0 & \text{if } \beta(x) < 0, \\ 1 & \text{if } \beta(x) > 0. \end{cases} \quad (12)$$

The left-hand side of Eq. 11 is a polynomial of degree at most two, and the right-hand side equals $-\beta$. Thus, we can write β in terms of its coefficients according to Eqs. 4 and 5. Equating coefficients in Eq. 11 yields $\nabla_r \mathcal{L} = -p$, which is exactly Eq. 7 with $\mu = 1$.

(1) If $p \neq 0$, then β is not the zero function and only vanishes on a measure-zero set. In consequence, χ^* is uniquely determined by Eq. 12 except on this measure-zero set, and agrees with χ as defined by Eq. 6 except on this set. Furthermore, we can replace p by $\bar{p} := p/\|p\| \in S^{n-1}$ and set $\mu = \|p\|$ in Eq. 7 in order to satisfy the conditions of Theorem 1. Application of this theorem shows that χ^* is a strict local optimizer for (RP).

(2) If $p = 0$, then $\nabla_r \mathcal{L} = 0$ according to Eq. 7. This equation implies the first-order optimality condition obtained from (RP) after modifying it to allow any $\chi : \Omega \rightarrow \mathbb{R}$, and thus χ^* is first-order optimal for this problem. \square

For many practical choices of f and g_1, \dots, g_k in (RP), we can exclude the case $p = 0$ a priori by inspecting the equation $\nabla_r \mathcal{L} = -p$, and showing that $\nabla_r \mathcal{L} = \nabla_r f - \sum_{i=1}^k \lambda_i \nabla_r g_i$ is nowhere zero. In particular, this is possible for *all* optimization problems considered in previous work, as we show in Section 4. This guarantees that any optimal solution to (RP) is binary, and thus also optimal for (TOP).

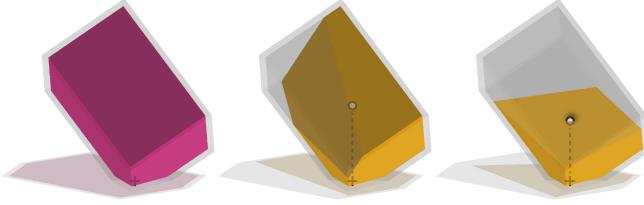


Fig. 6. Static Stability. The design domain (pink) is enclosed in a fixed solid shell (transparent). The solutions with maximal mass (*center*) and lowest center of mass (*right*) both have a planar interface, and their centers of mass (o) project downwards onto the target point (+).

Even if $p = 0$ is not excluded a priori, then finding a numerical solution with $p \in S^{n-1}$ shows optimality by virtue of Theorem 1. In Section 6.6, we show an academic example of a problem with solutions that satisfy $p = 0$, and argue that this will only occur for problems that are ill-posed.

4 OPTIMIZATION PROBLEMS

We give an overview of useful optimization problems that can be cast in the form of (TOP), with pointers to the works that introduced them. Our optimality conditions shine a new light on the geometric structure of these problems by guaranteeing a quadric interface of the optimal mass distribution. For some problems, we can even clarify the Euclidean type of the quadric and obtain more specialized knowledge prior to computing a numerical solution.

4.1 Mass Moments

Many problems are formulated more readily in terms of auxiliary quantities, such as the mass moments used in the introduction, rather than the raw moments v , ℓ , and Q . We provide a reference of these quantities here and use them in the remainder of this section.

Assuming a constant density $\varrho > 0$, the mass m , center of mass c , and inertia tensor I^0 with respect to the origin are given by

$$m = \varrho v, \quad c = \ell/v, \quad \text{and} \quad I^0 = \varrho \cdot (\text{tr } Q \cdot \text{id}_{d \times d} - Q),$$

where $\text{id}_{d \times d} \in \mathbb{R}^{d \times d}$ denotes the identity matrix. The inertia tensor I^c with respect to the center of mass is given by

$$I^c = I^0 - m \cdot (\langle c, c \rangle \cdot \text{id}_{d \times d} - c \otimes c).$$

Applications also benefit from the ability to prescribe part of the object as solid. This is useful, e.g., to add a solid shell around Ω to determine the outward appearance and conceal any cavities. To account for this additional fixed material, one can precompute the raw moment vector of the shell at the beginning and add it to r before using it in further computations. This leaves all results and formulas unchanged because it is only a constant offset.

As gravitational acceleration plays a role in many of the problems described below, we fix the convention that gravity acts in the “vertical” direction $-e_3 \in \mathbb{R}^3$, while e_1 and e_2 span a “horizontal” plane, perpendicular to gravity.

4.2 Static Stability

An object placed on a flat horizontal surface will be in static equilibrium if its center of mass projects downwards into the convex

hull of all points that are in contact with the surface. The problem introduced by Prévost et al. [2013] is to find a mass distribution that achieves this by constraining the first and second coordinates of the center of mass to a desired location, usually the center of the largest incircle of the convex hull of contact points. The objective function may be chosen to further increase the stability of the position.

We choose our coordinate system such that the origin coincides with the desired values of c_1 and c_2 . This simplifies the equality constraints to $c_1 = 0 = c_2$, which is equivalent to $\ell_1 = 0 = \ell_2$.

Maximize mass. One simple optimization objective that increases the static stability of the object with respect to external forces is to maximize its mass, or equivalently, its volume,

$$\min -v \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2.$$

Writing Eq. 7 in coordinates for this problem gives

$$(1, \lambda_1, \lambda_2, 0, 0, \dots, 0)^\top = \mu \cdot (a, b_1, b_2, b_3, A_{11}, \dots, A_{12})^\top$$

with $\mu > 0$. Note that because only the raw moments of up to first order appear in this problem, all entries of A in Eq. 4 vanish. Thus β is an affine function, and the optimal solution will be given by a *planar interface* between the empty and solid region. Using Eq. 6, we see that the optimal mass distribution satisfies $\chi^*(x) = 1$ for exactly those $x \in \Omega$ such that $1 + \lambda_1 x_1 + \lambda_2 x_2 > 0$, for some choice of $\lambda_1, \lambda_2 \in \mathbb{R}$. Thus, the interface is a *vertical plane*, and the halfspace containing the origin is the solid region.

Lower center of mass. Another objective to increase stability is to lower the center of mass as much as possible:

$$\min c_3 \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2.$$

The optimality conditions for this problem are

$$(c_3/v, \lambda_1, \lambda_2, -1/v, 0, \dots, 0) = \mu \cdot (a, b_1, b_2, b_3, A_{11}, \dots, A_{12}). \quad (13)$$

This shows that the interface of the optimal mass distribution is a *non-vertical plane*, and the halfspace below this plane is the solid region. This follows from $b_3 = -1/(\mu v) < 0$. Moreover, we can show that this plane contains the center of mass by verifying $\beta(c) = 0$.

Note that in both versions of the static stability problem, it was easy to see that $p \neq 0$, because we had $a = 1/\mu > 0$ in the first, and $b_3 < 0$ in the second. Thus, Theorem 2 guarantees that all local optimizers are strict, and of the types described here. Fig. 6 illustrates both variants of the problem on a geometrically simple example.

4.3 Buoyant Stability

A problem described by Wang and Whiting [2016] is to optimize a three-dimensional object such that it stably floats when partially submerged in a liquid in a prescribed orientation and at a prescribed liquid level, assumed to coincide with $\{x \in \mathbb{R}^3 : \langle x, e_3 \rangle = 0\}$.

The conditions for mechanical equilibrium and stability depend on $\Omega_{\text{displ}} \subset \mathbb{R}^3$, the domain of the liquid displaced by the object in the prescribed floating pose. We follow Wang and Whiting in assuming that there is a fixed solid shell Ω_{shell} around the optimization domain Ω , such that $\Omega_{\text{displ}} = (\Omega \cup \Omega_{\text{shell}}) \cap \{x \in \mathbb{R}^3 : \langle x, e_3 \rangle \leq 0\}$ is known ahead of time. This allows us to compute the volume $v_{\text{liquid}} > 0$ of the displaced liquid, and its center of mass c_b , known as

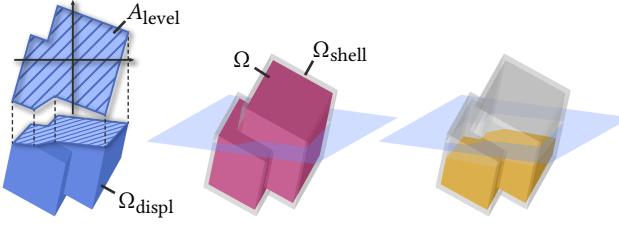


Fig. 7. Buoyant Stability. *Center:* Target configuration with design domain Ω (pink) enclosed by fixed solid shell Ω_{shell} (transparent) and the prescribed waterline. *Left:* Volume of displaced liquid Ω_{displ} (blue) and its top surface A_{level} (striped). *Right:* Optimal mass distribution (yellow).

the *center of buoyancy*. After choosing coordinates so $c_{b,1} = 0 = c_{b,2}$, the conditions for equilibrium are

$$\varrho \cdot (v + v_{\text{shell}}) = \varrho_{\text{liquid}} \cdot v_{\text{liquid}}, \quad \text{and} \quad c_1 = 0 = c_2,$$

where $v_{\text{shell}} > 0$ is the volume of the solid shell, and $\varrho_{\text{liquid}} > 0$ is the density of the liquid.

For the floating pose to be retained in the real world, one has to guarantee stability of the equilibrium under perturbations. The condition $c_3 < c_{b,3}$ is sufficient, but impractical to enforce and fortunately not necessary. A weaker sufficient condition is $c_3 < z_m$, with z_m the height of the so-called *metacenter*. This condition takes into the account the stabilizing moment produced by changes of Ω_{displ} under tilting motions of the floating object. The metacenter depends on the *level area*, the two-dimensional domain

$$A_{\text{level}} = \{(x, y)^T \in \mathbb{R}^2 : (x, y, 0)^T \in \Omega_{\text{displ}}\},$$

which represents the top surface of the displaced liquid volume. We can compute the raw moments of A_{level} from Eq. 2 with $d = 2$, and its second-order area tensor $I_{\text{level}}^c \in \mathbb{R}^{2 \times 2}$ with respect to the area center from the formulas in Section 4.1 (setting $\varrho = 1$ in this calculation). Finally, the height of the metacenter is $z_m = c_{b,3} + \lambda_{\min}/v_{\text{liquid}}$, where $\lambda_{\min} > 0$ is the smallest eigenvalue of I_{level}^c .

To test whether the problem is feasible, we solve

$$\min c_3 \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2, \quad v = (\varrho_{\text{liquid}}/\varrho) \cdot v_{\text{liquid}} - v_{\text{shell}}.$$

If the optimal solution satisfies $c_3 < z_m$, the equilibrium is stable; otherwise, the problem is infeasible. The optimality conditions are the same as Eq. 13 except for the first component, which is $c_3/v + \lambda_3 = \mu \cdot a$ in this problem. Qualitatively, this does not change the solution type, which has a *non-vertical, planar* interface, with the halfspace below the plane solid. Fig. 7 shows an example of a floating body, with an illustration of all the domains relevant for computations.

4.4 Fidget Spinners

A fidget spinner is a flat mechanism optimized to stably rotate about its center of mass with high angular momentum. We can achieve this through optimization by finding a mass distribution on $\Omega \subset \mathbb{R}^2$ that maximizes the polar moment of inertia. Assuming that the origin coincides with the desired center of mass, this yields

$$\min -\text{tr } I^o \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2,$$

where we note that $\text{tr } I^o = \varrho \cdot (Q_{11} + Q_{22})$. Eq. 7 gives

$$(0, \lambda_1, \lambda_2, \varrho, \varrho, 0)^T = \mu \cdot (a, b_1, b_2, A_{11}, A_{22}, A_{12})^T,$$

and Eq. 6 shows that the solid region contains exactly those $x \in \Omega$ such that $\langle x + \lambda/\varrho, x \rangle > 0$, with $\lambda = (\lambda_1, \lambda_2)^T$. This shows that the optimal interface is a *circle that intersects the origin*, having center $-\lambda/(\varrho)$. The empty region is the disk bounded by this circle.

4.5 Yo-yos

A yo-yo is a toy that can stably spin about a given axis when suspended from its center of mass. For this example only, we assume that coordinates axes are chosen to align e_3 with the desired spinning axis, rather than the direction of gravity; this is to keep notation consistent with the section on spinning tops, below. A stable spin is possible only if e_3 is an eigenvector of I^c and corresponds either to the minimal or maximal eigenvalue. As such, I^c needs to be block-diagonal with blocks spanning indices $\{1, 2\}$ and $\{3\}$. To maximize angular momentum, Bächer et al. [2014] suggest to maximize the eigenvalue of e_3 relative to the others and arrive at

$$\min f_{\text{yo}} := \frac{(I_{11}^c)^2 + (I_{22}^c)^2 + 2(I_{12}^c)^2}{(I_{33}^c)^2} \quad \text{s.t.} \quad \ell = 0, \quad I_{13}^c = 0 = I_{23}^c,$$

where the numerator of the objective is the sum of squared eigenvalues of the $\{1, 2\}$ -block of I^c —which equals I^o at any feasible point due to the constraint on ℓ . The optimality conditions are

$$\begin{aligned} (0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)^T &= \mu \cdot (a, b_1, b_2, b_3, A_{13}, A_{23})^T, \\ 2/(I_{33}^o)^2 \cdot (f_{\text{yo}} I_{33}^o - I_{22}^o, f_{\text{yo}} I_{33}^o - I_{11}^o, -(I_{11}^o + I_{22}^o), f_{\text{yo}} I_{33}^o + 2I_{12}^o)^T \\ &= \mu \cdot (A_{11}, A_{22}, A_{33}, A_{12})^T, \end{aligned}$$

and we can deduce from $a = 0$ and $A_{33} < 0$ that the optimal interface intersects the origin, and that any line parallel to the spinning axis e_3 will be contained in the empty region, except for at most a segment of finite length. It also follows that $p \neq 0$, so the problem is not degenerate for any domain.

4.6 Spinning Tops

Bächer et al. [2014] model spinning tops in a way similar to yo-yos, but with the spinning axis aligned to gravity, and the center of mass allowed to move along this axis during optimization. They also suggest adding a secondary objective to lower the center of mass, which yields

$$\min f_{\text{yo}} + w_c \cdot c_3 \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2, \quad I_{13}^c = 0 = I_{23}^c,$$

with the optimization weight w_c having units of reciprocal length. Because c is free to move along e_3 , we have

$$I^c = I^o - mc_3^2 \cdot (e_1 \otimes e_1 + e_2 \otimes e_2),$$

at any feasible point instead of $I^c = I^o$, slightly complicating the optimality conditions. However, $A_{33} = -2(I_{11}^c + I_{22}^c)/[\mu(I_{33}^c)^2] < 0$ still holds, so $p \neq 0$ is guaranteed.

5 NUMERICAL METHOD

To discover local optimizers of (TOP) numerically, we need to find $p \in S^{n-1}$ such that the mass distribution $\chi : \Omega \rightarrow \{0, 1\}$ as defined by Eqs. 4–6 satisfies Eq. 7 and $0 = g := (g_1, \dots, g_k)^T$. As demonstrated in the previous section, the expressions for the entries of $\nabla_r \mathcal{L} = \nabla_r f - \sum_{i=1}^k \lambda_i \nabla_r g_i$ tend to be simple for most practical problems, but they depend on the raw moments $r \in \mathbb{R}^n$. These

quantities arise from integrating monomials of up to second order over $\omega = \{x \in \Omega : \beta(x) > 0\}$, a task which we discuss in Section 5.2 under the assumption that the boundary $\partial\Omega$ of Ω is triangulated.

5.1 Solving the Optimality Conditions

We apply a variant of Newton's method in order to find solutions to the non-linear equation system

$$\nabla_r f(r(p)) = -\mu p + \text{Jac}_r g(r(p))^\top \lambda, \quad g(r(p)) = 0, \quad \|p\|^2 = 1, \quad (14)$$

in the unknowns $p \in \mathbb{R}^n$, $\lambda := (\lambda_1, \dots, \lambda_k)^\top \in \mathbb{R}^k$, and $\mu > 0$. Here, $\text{Jac}_r g(r(p)) \in \mathbb{R}^{k \times n}$ denotes the Jacobian matrix of g .

Our numerical method starts with an initial guess $p^0 \in S^{n-1}$ and then computes iterates p^1, p^2, \dots until the ℓ_2 -error of Eq. 14 is below a termination threshold, which we set to 10^{-12} . We update p by tracing geodesic arcs on S^{n-1} , so $\|p^j\|^2 = 1$ is satisfied by construction for all $j \geq 0$. The unknowns λ and μ are not updated according to Newton's method, but recalculated in every step to minimize the ℓ_2 -error of the left-most equation in Eq. 14. This is done by computing the orthogonal projection of $\nabla_r f$ onto $\text{span}\{-p, \nabla_r g_1, \dots, \nabla_r g_k\}$ and taking its coefficients.

Algorithm. We compute one iteration as follows: Given $p^j \in S^{n-1}$, find the corresponding λ^j and μ^j according to

$$(\lambda^j, \mu^j) = \arg \min_{\lambda, \mu} \|\nabla_r f(r(p^j)) + \mu p^j - \text{Jac}_r g(r(p^j))^\top \lambda\|^2,$$

by solving a linear system. Then, we compute the residuals

$$e_f^j = \nabla_r f(r(p^j)) + \mu^j p^j - \text{Jac}_r g(r(p^j))^\top \lambda^j, \quad e_g^j = g(r(p^j)),$$

and, omitting the argument $r(p^j)$, the Newton step Δp^j from

$$\begin{pmatrix} \text{Hess}_r \mathcal{L}^j \cdot \text{Jac}_p r + \mu^j \cdot \text{id}_{n \times n} & -\text{Jac}_r g^\top & p^j \\ \text{Jac}_r g \cdot \text{Jac}_p r & 0_{k \times k} & 0_{k \times 1} \\ 2(p^j)^\top & 0_{1 \times k} & 0_{1 \times 1} \end{pmatrix} \begin{pmatrix} \Delta p^j \\ \sim \\ \sim \end{pmatrix} = \begin{pmatrix} -e_f^j \\ -e_g^j \\ 0 \end{pmatrix},$$

with $\mathcal{L}^j = f(r(p^j)) - \langle g(r(p^j)), \lambda^j \rangle$, and Hess denoting the Hessian matrix. Note that $\text{Hess}_r \mathcal{L}^j \cdot \text{Jac}_p r = \text{Jac}_p \nabla_r \mathcal{L}^j$.

The next iterate is computed as

$$p^{j+1} = \cos t^j \cdot p^j + \sin t^j \cdot \Delta p^j / \|\Delta p^j\|,$$

which lies on the geodesic arc emanating from p^j in direction Δp^j on S^{n-1} . The step size $t^j > 0$ is initialized with $\min\{\|\Delta p^j\|, t_{\max}\}$ and then iteratively reduced by a factor of $\alpha \in (0, 1)$ until the ℓ_2 -error of Eq. 14 is smaller at p^{j+1} than at p^j . The choice $t^j = \|\Delta p^j\|$ produces a true Newton step, but outside basins of attraction, this may result in large jumps. In particular, $t^j = \pi$ implies $p^{j+1} = -p^j$, which inverts the mass distribution, and is clearly too large a step. Thus t_{\max} should be chosen well below π . In our implementation, we set $\alpha = 1/2 = t_{\max}$, but the convergence speed is insensitive to the exact values.

Initial Guess. We draw $p^0 \in S^{n-1}$ in such a way that it respects the properties that can be derived a priori for the specific instance of (TOP) as discussed in Section 4, e.g., $a = 0$ and $A_{33} < 0$ for the yo-yo problem. An initial guess is accepted if $\mu^0 > 0$ and $\text{Jac}_p r(p^0) \neq 0_{n \times n}$.

The inequality encourages the outcome that $\mu > 0$ still holds after convergence, corresponding to the discovery of a local minimizer. The non-vanishing of $\text{Jac}_p r$ avoids starting the optimization in a

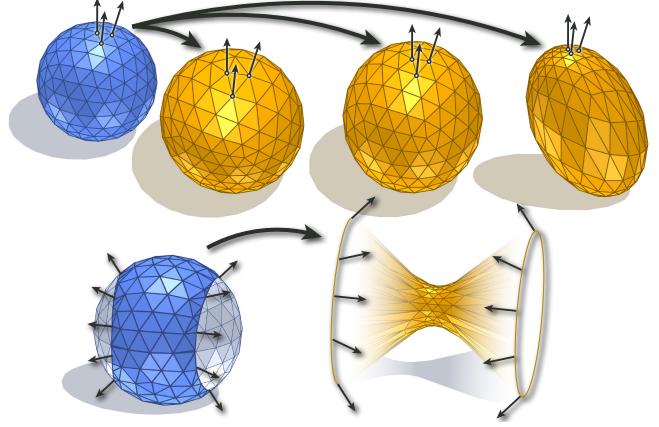


Fig. 8. **Quadric Triangulation.** *Top:* Uniform triangulation of a sphere (blue) is mapped to different ellipsoids in such a way that each vertex keeps its original normal. *Bottom:* Only vertices in the blue region correspond to valid surface normals on the hyperboloid, so the gray region of the sphere is truncated. Normals at the boundary of the blue region (left) are mapped to ideal normals “at infinity” (right).

state where the interface $\{x \in \mathbb{R}^d : \beta(x) = 0\}$ does not intersect Ω , which would zero out all sensitivities with respect to p . We run the iteration starting from a randomly drawn initial guess that meets these conditions and check if $\mu > 0$ holds after convergence. If so, the algorithm terminates; otherwise, the iteration is restarted from another random initial guess. For the optimization problems considered in Section 4, it is rare that any restarts are needed.

5.2 Moment Computation

Most of the computation time in running the numerical iteration goes into computing $r(p)$ and $\text{Jac}_p r(p)$, the raw moments defined in Eq. 2 and their derivatives. The computation of r amounts to integrating monomials over the domain $\omega = \Omega \cap Q^+$, where $Q^+ := \{x \in \mathbb{R}^d : \beta(x) > 0\}$. Assuming that the boundary $\partial\Omega$ is a triangulated surface, $\partial\omega$ will generally consist of a triangulated portion inherited from $\partial\Omega$ and a smooth portion which is contained in the surface $Q := \{x \in \mathbb{R}^d : \beta(x) = 0\}$. In dimension three, Q is either an ellipsoid, hyperboloid, paraboloid, or a degenerate quadric.

In theory, these integrals can be solved analytically by computing the conic sections resulting from $\partial\Omega \cap Q$ explicitly, and applying Stokes' theorem to reduce the volume integrals to surface integrals, and finally to curve integrals. However, this involves case distinctions depending on the types of the quadric and conic sections, and many algebraic manipulations to arrive at a closed-form result.

Below, we describe a different approach, in which we triangulate the smooth quadric Q such that the vertex positions are differentiable in p . This simplifies the problem by letting us use a triangle mesh Boolean between $\partial\Omega$ and the discretized quadric to produce a triangulation of $\partial\omega$. This in turn allows the computation of r by applying the divergence theorem once and using triangle quadrature rules that integrate polynomials of the required degree exactly.

Quadratic Triangulation. We propose a triangulation of non-degenerate quadrics whose vertex density is adapted to (*Gaussian*) curvature by construction. The algorithm is based on the *Gauss map* $v : Q \rightarrow S^2 \subset \mathbb{R}^3$, which assigns to every point $x \in Q$ the oriented unit normal vector $v(x)$. On any surface, this map has the property that its Jacobian determinant equals curvature; on non-degenerate quadrics, v is also injective. Thus, applying the inverse Gauss map $v^{-1} : v(Q) \rightarrow Q$ to a uniformly distributed point set on the image $v(Q)$ yields a point set on Q whose density is directly proportional to curvature.

We use this idea for triangulation by fixing a uniform-density discretization of the unit sphere S^2 , such as an *icosphere*. If Q is an ellipsoid, then v is bijective, and we can triangulate Q simply by applying v^{-1} to the vertices of the icosphere. If Q is a hyperboloid, then v is not surjective, and not all vertices of the icosphere will have a preimage under v . In this case, we only apply v^{-1} to the icosphere vertices contained in $v(Q)$, and add vertices at the intersections of icosphere edges and the boundary of $v(Q)$. These newly created vertices correspond to *ideal points* of Q , which can be thought of as points at infinity. Finally, to arrive at a bounded discretization, we truncate our triangulation of Q at a radius large enough so as not to affect the result of the intersection between Q^+ and Ω .

All vertex positions, including those of ideal vertices, are differentiable in p , as evidenced by the formulas in the supplemental document. The mapping procedure is shown in Fig. 8 on ellipsoids (top), and a hyperboloid (bottom).

6 RESULTS

6.1 Implementation and Fabrication Details

We implemented our numerical method as a single-threaded Matlab application that calls the mesh Boolean code of Cherchi et al. [2022] in order to compute the intersection between the problem domain and a triangulated quadric in every iteration. Our examples have all been modeled in the CAD system *Onshape*, and consist of a solid shell, which remains fixed during optimization, and the domain.

To run our code, we temporarily convert shell and domain to a triangle mesh representation, and uniformly rescale it to have approximately unit length. This is to make the convergence threshold comparable between different examples. After our numerical method finds the optimal mass distribution in the form of a quadric surface, we use a CAD script to construct this quadric inside *Onshape* and incorporate the mass distribution into the model. Then we prepare the model for fabrication by splitting it up into geometrically simpler pieces in order to reduce support material, and to prevent cavities that would trap support material during the print.

All of our samples are 3d-printed from PLA, using PVA as a support material, on an *Ultimaker S7 Pro* 3d printer, set to a precision of 0.1 mm layer height. We use a solid fill-in of 100% in order to mimic a homogeneous constant-density material as accurately as possible. For the examples involving buoyant stability, the relative density of the solid material and water enters the computation. To estimate it, we printed and weighed a 2 cm × 2 cm × 2 cm sample cube, which yielded $\rho = 1187 \text{ kg/m}^3$ for printed PLA.

6.2 Baseline Examples

Figs. 1 and 9 show all of our fabricated examples in three stages: the initial CAD geometry consisting of the fixed shell (transparent gray) and the design domain (pink); the optimized mass distribution (yellow) in a cut-off view; and the 3d-printed prototype. The supplemental video also shows the 3d-printed parts before assembly, as well as the finished prototype in action.

Static Stability. We designed a vase with several topological handles that leans to one side and is balanced on a small cut-off portion of the bottom edge. The fixed material takes the form of a double-walled shell, see Fig. 9 (row 1, gray), which encloses the design domain. We use mass maximization as the optimization objective, which leads to an optimal mass distribution with a vertical planar interface. To prepare the model for printing, we separate the filled and hollow portions of the vase, and furthermore split the hollow portion between the two walls, as seen in the supplemental video.

Buoyant Stability. We tested this application on two different models. The first is a tea infuser, consisting of a hollow duck-shaped torus with a handle, and a submerged strainer containing tea leaves. The prescribed waterline is chosen so the torus is about two-thirds underwater. The optimal solution partially fills the torus with material, in such a way that there is more material on the side opposite the handle, in order to balance the weight, see Fig. 9 (row 2).

The second model is a floating platform consisting of two parts. Part A has a cat on one side, and a dog on the other. It is optimized such that it only floats stably if the cat is pointing upwards, and the dog is submerged. Part B, which has a person standing on one side, can be attached to part A such that the person stands next to the dog. In this configuration, the assembled platform floats only if the dog is pointing upwards, and the cat is submerged, as seen in Fig. 9 (row 3). To realize this example, we first optimized part A separately, and then used the optimized solution as part of the fixed shell when optimizing part B.

Spinning Tops. The spinning top shown in Fig. 1 consists of an asymmetrically placed heart on a pin that acts as the spinning axis and is inspired by an example of Bächer et al. [2014]. The optimal mass distribution is the result of balancing two competing objectives that try to place mass away from the spinning axis while simultaneously lowering the center of mass. This results in a material-air interface shaped like a one-sheeted hyperboloid.

The second spinning top, shown in Fig. 9 (row 6) is a stress test in terms of geometric complexity. Here, the solid shell is formed by an array of curved tendrils extending out from the center of the spinning top. The design domain is an ellipsoid pierced by these tendrils, resulting in a volume with many thin, long holes and high geometric complexity. Our numerical method finds that the optimal interface is a two-sheeted hyperboloid, resulting in a spinning top that allows for long and stable spins as shown in the video.

Yo-yo. The yo-yo optimization problem is similar to that of the spinning top, and tends to result in similarly shaped optimal mass distributions. Fig. 9 (row 5) shows an example of a yo-yo modeled after a jellyfish, for which the optimal interface was found to be a



Fig. 9. Optimized Examples. The left column depicts the fixed outer shell in transparent-grey, the design domain in pink, and for the buoyancy examples, the target water line in blue. The middle column shows a cut-away view of the optimized mass distribution in yellow (see Section 6 for the detailed description) and the fabricated objects are represented in the photographs in the right column.

ACM Trans. Graph., Vol. 43, No. 4, Article 78. Publication date: July 2024.

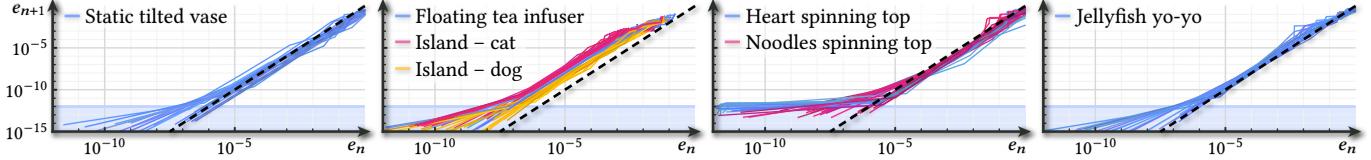


Fig. 10. Quadratic Convergence. Convergence plots for our numerical method, grouped by application. From left to right: Static stability, buoyant stability, spinning tops, yo-yo. We plot the ℓ_2 -error of Eq. 14 at iteration n against that at iteration $n + 1$, for all successful runs used to generate the data in Table 1. A graph parallel to the dashed line indicates quadratic convergence. The shaded region at the bottom contains points below the convergence threshold.

two-sheeted hyperboloid. The supplemental video shows that the yo-yo can run stably up and down a string without losing balance.

Fidget Spinner. For this 2d example, we represent the domain geometry as a set of edge loops that form the boundary of a flower design with many topological holes. The center of mass does not initially coincide with the center of the flower because of the asymmetric hole placement. We design the fidget spinner by extruding three layers: The ones at the top and bottom are fixed parts of the shell, and the center part contains the design domain, enclosed by a solid rim on the outside and a solid ring at the center to house the ball bearing, as shown in Fig. 9 (row 4, left). Despite being three-dimensional in this illustration, the relevant mass moments can be computed from the 2d domain and the shell parts, so the example is computationally 2d. The optimal solution cuts away a circular region from the design domain in such a way that the center of mass of the leftover material coincides with the center of the ball bearing.

6.3 Convergence and Computation Time

For each baseline example, we ran our numerical method 50 times from randomly chosen initial points, which are pruned according to the heuristic discussed in Section 5.1. In Table 1, we record statistics for each example, such as the geometric complexity, average number of random restarts necessary to find a minimum, the number of

Newton steps until convergence, number of function evaluations, and computation time. Fig. 10 shows the trajectory of the ℓ_2 -error and clearly indicates quadratic convergence across all applications.

The median computation time until convergence is split into the time that goes into computing the residual vector of Eq. 14, and the time spent on computing derivatives for the Newton step. The former contains the mesh Boolean operation as well as the computation of mass moments on the resulting triangle mesh.

For most attempts, our numerical method converges from the first random initial point, as evidenced by a number close to 1.00 in the “Avg. restarts” column of Table 1. This number is highest for the heart spinning top example, for which the algorithm needed to try a second initial point in 28% of all runs to achieve convergence. The attempts that did not converge were interrupted after 60 function evaluations. The only other failure mode was convergence to a local maximum instead of a minimum, which happened in 2/50 attempts for each of the spinning tops, and 1/50 attempts for the yo-yo, also causing a restart.

6.4 Pareto Exploration

In Section 4.2, we described two possible objective functions to improve the static stability of an object, based on maximizing the volume and lowering the center of mass. We can explore the trade-off between the two criteria by considering a weighted objective

Table 1. Performance. We ran our numerical method for each example from a set of randomly drawn initial points, and report the statistics of the Newton solve and computation times across 50 runs. *Avg. restarts*: Average number of restarts from random initial points until numerical method converges (1 = convergence on first attempt). We restart if 60 function evaluations have been reached without convergence. *Med. Newton iterations*: The median number of Newton iterations to convergence among all successful runs. *Med. function evaluations*: The median number of function evaluations among all successful runs. This also counts function evaluations due to back-propagation steps. *Med. computation time*: The median computation time until convergence, split between the time spent on evaluating the residual vector, which includes the mesh Boolean operation and mass moment computation, and the derivatives for the Newton step. [†]For the 2d fidget spinner example only, we used Matlab’s fsolve method with the trust-region-dogleg algorithm instead of our own method. Due to the code organization, the computation time is only available as a whole. Here, we report #edges instead of #faces.

Example	#vertices / #faces	Avg. restarts	Med. Newton iterations	Med. function evaluations	Med. computation time (s)	
					Residual	Derivatives
Static tilted vase	23k / 46k	1.00	7	8	1.97	2.26
Floating tea infuser	8.1k / 16k	1.00	8	10	0.51	0.30
Floating island (Cat side)	204 / 408	1.02	11	15	0.23	0.04
Floating island (Dog+human side)	1.3k / 2.6k	1.00	7	7	1.52	0.15
Heart spinning top	5.2k / 10k	1.28	14	16	1.30	0.81
Noodles spinning top	60k / 120k	1.06	11	13	6.93	7.40
Jellyfish yo-yo	14k / 27k	1.02	8	9	1.08	0.81
Flower fidget spinner[†]	4.7k / 4.7k	1.13	11	12	0.53	-

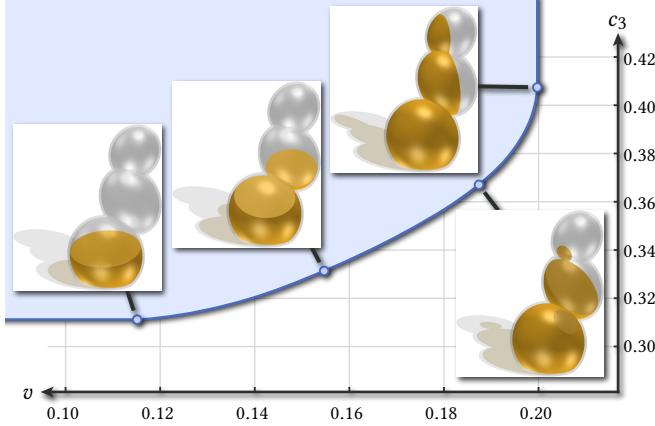


Fig. 11. Pareto Front Static Stability. Pareto front (blue) of the two objectives for the static stability problem. Four particular solutions (yellow) of Pareto points shown, from maximizing the volume v only (*top-most*) to minimizing c_3 only (*left-most*).

that combines the two, leading to

$$\min -\kappa \cdot v + (1 - \kappa) \cdot c_3, \quad \text{s.t.} \quad \ell_1 = 0 = \ell_2,$$

for a weight $\kappa \in [0, 1]$. Swiping the whole range of κ , and plotting the two objective values $(v, c_3) \in \mathbb{R}^2$ for each optimal solution, we recover the *Pareto front* of the problem. This gives rise to a range of mass distributions that interpolate between the extremal solutions attained for $\kappa = 0$ and $\kappa = 1$. The Pareto front, as well as four particular solutions are shown in Fig. 11 on a model of three stacked eggs, inspired by an example of Prévost et al. [2013].

Similarly, the spinning top problem from Section 4.6 incorporates an extra weighted term to enhance the rotational stability of the object by lowering the center of mass. By swiping the range of weights and plotting the points $(f_{yo}, c_3) \in \mathbb{R}^2$ corresponding to each optimal solution, we obtain the Pareto front of this problem. Fig. 12 shows this front along with four particular solutions of the heart spinning top example, with the render at the bottom-left corresponding to the physical prototype shown in Fig. 1. Note that the Euclidean type of the quadric changes as we move along the Pareto front: It starts as a one-sheeted hyperboloid at the top of the graph, transitions to a two-sheeted hyperboloid near the center, and back to a one-sheeted hyperboloid near the right-most point.

If a point on the Pareto front has already been computed, then adjusting the solution to compute a nearby point takes less than a second on either of these examples with our numerical method. This allows exploration of the Pareto front at an interactive rate.

6.5 Problem Feasibility

It is possible to choose the constraints g_1, \dots, g_k and the domain Ω in (TOP) such that no feasible solution exists. For example, this will occur in the static stability problem, see Section 4.2, if the target point lies outside the convex hull of $\Omega \cup \Omega_{\text{shell}}$, projected onto the ground plane. In other applications, such as the spinning top problem, it is often not obvious from visual inspection whether a given domain and shell geometry will admit a feasible solution.

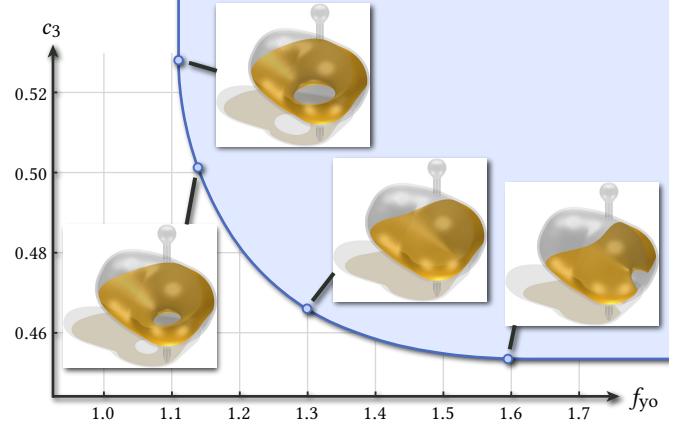


Fig. 12. Pareto Front Spinning Top. Pareto front (blue) of the two competing objectives for the spinning top problem. Four particular solutions (yellow) of Pareto points shown, from minimizing f_{yo} only (*top-most*) to minimizing c_3 only (*right-most*).

We investigate the behavior of our numerical method for problems with a very small feasible region. To this end, we use the input geometry shown in Fig. 13 as input to the spinning top problem. The geometry consists of a tilted cube on a pin and is parametrized by the distance $d > 0$ between the cube center and the centerline of the pin, which acts as the spinning axis. Beyond a certain critical distance $d_{\text{crit}} \approx 1$, which we determined through binary search to an accuracy of nine significant digits, the problem becomes infeasible.

We tested convergence of our algorithm by setting d to

$$d_m := d_{\text{crit}} - 10^{-m}, \quad \text{for } m = 1, \dots, 9,$$

and plotting the ℓ_2 -error of Eq. 14 as a function of the number of Newton iterations in Fig. 13. For high values of m , when the feasible region is very small, our numerical method spends more iterations in the linear regime, but once an ℓ_2 -error of about 10^{-5} is reached, quadratic convergence is obtained even for $m = 9$. To make iteration counts comparable, we started each test from the solution to the unperturbed geometry, shown in Fig. 13 (*bottom-left*).

6.6 Ill-Posed Problems

In Section 3.4, we pointed out the existence of the “degenerate” case $p = 0$, which we excluded for all practical applications described in Section 4. However, we can construct academic examples that show under which conditions $p = 0$ occurs.

Consider the domain $\Omega = (0, 1) \subset \mathbb{R}$ with the problem

$$\min_{\chi: \Omega \rightarrow [0, 1]} (v(\chi) - 1/2)^2.$$

Naturally, any function χ on Ω with an area of $1/2$ below its graph will have a vanishing objective value and is thus optimal. However, there is an uncountably infinite number of such optimal choices, not all of which are binary. The issue with this example is that it is really a constraint-satisfaction problem ($v = 1/2$) in the guise of an optimization problem. If the constraint can be satisfied, as is the case here, this will cause $\nabla f = 0$ to hold, and in consequence $p = -\nabla f = 0$. A problem of this kind usually has infinitely many

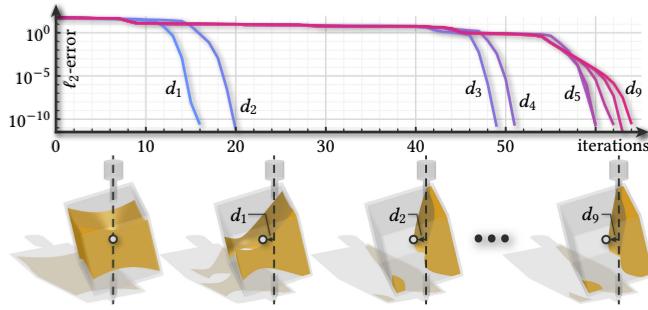


Fig. 13. Feasibility Test. We run our numerical method on a family of input meshes, where the mesh labeled d_m is within 10^{-m} units of the infeasible region. The closer a mesh is to being infeasible, the more iterations our method spends in the linear regime before achieving quadratic convergence.

solutions whenever it is feasible, because χ possesses infinitely many degrees of freedom, but there are only finitely many constraints.

A slight variation of this problem can be obtained by matching the size of the domain to the objective function in such a way that there is only one optimal solution, which is obtained either when the domain is completely empty or filled. To see this, consider the same problem as before, but with domain $\Omega = (0, 1/2)$. Now, the only function that achieves $v = 1/2$ is $\chi \equiv 1$, and $p = 0$ still holds.

These two problem types—infinitely many optimal solutions, or a single solution that is obtained either by $\chi \equiv 0$ or $\chi \equiv 1$ —are the only two instances of $p = 0$ that we have found. Proving that no others exist is an avenue for future research.

7 CONCLUSION

In this work, we analyze the class of topology optimization problems that depend only on the mass moments of the object under consideration. This type of problem has been studied for a variety of applications in previous work, ranging from static and buoyant stability to the design of spinning tops and yo-yos. We show that the optimal solutions to all these problems are binary mass distributions in which the material-air interface is formed by a quadric, regardless of the shape of the problem domain. This allows us to replace the traditional voxel-based topology optimization approach with a small system of equations that can be solved numerically to compute the coefficients of the quadric.

A current limitation of our method is that the computation of mass moments requires a detour via triangle meshes, even though the result is always a smooth surface. It would be interesting to implement our method directly in a differentiable CAD kernel in order to compute intersections between the domain and a quadric directly based on parametric surfaces, and drive the numerical method with derivatives obtained from the kernel.

Despite already covering a range of applications in its current form, the algorithm we describe is less flexible than some methods from previous work that apply cage deformations in addition to hollowing. This can resolve some problem instances for which hollowing alone is insufficient. Adapting our approach to optimize

the outer geometry as well as the mass distribution is a promising direction to increase its applicability.

ACKNOWLEDGMENTS

We thank Gianmarco Cherchi for his help in tailoring the Mesh Booleans code for this project, Stefan Jeschke for his help with the photographs, Irina-Malina Strugaru and Aleksei Kalinov for their help with the samples, and the anonymous reviewers as well as the members of the ISTA Visual Computing Group for their feedback. This project was funded in part by the European Research Council (ERC Consolidator Grant 101045083 CoDiNA).

REFERENCES

- Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. 2014. Spin-It: Optimizing Moment of Inertia for Spinnable Objects. *ACM Trans. Graph.* 33, 4, Article 96 (jul 2014), 10 pages. <https://doi.org/10.1145/2601097.2601157>
- Martin Philip Bendsoe and Ole Sigmund. 2003. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.
- Gianmarco Cherchi, Fabio Pellacini, Marco Attene, and Marco Livesu. 2022. Interactive and Robust Mesh Booleans. *ACM Trans. Graph.* 41, 6, Article 248 (nov 2022), 14 pages. <https://doi.org/10.1145/3550454.3555460>
- Joshua D. Deaton and Ramana V. Grandhi. 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization* 49 (2014), 1–38. <https://api.semanticscholar.org/CorpusID:122755935>
- Jiawei Feng, Jianzhong Fu, Zhiwei Lin, Ce Shang, and Bin Li. 2018. A review of the design methods of complex topology structures for 3D printing. *Visual Computing for Industry, Biomedicine, and Art* 1, 1 (2018), 1–16.
- Alexandra Ion, Johannes Froehnhofer, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial Mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (*UIST ’16*). Association for Computing Machinery, New York, NY, USA, 529–539. <https://doi.org/10.1145/2984511.2984540>
- Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-last: strength to weight 3D printed objects. *ACM Trans. Graph.* 33, 4, Article 97 (jul 2014), 10 pages. <https://doi.org/10.1145/2601097.2601168>
- Przemyslaw Musalski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2015. Reduced-Order Shape Optimization Using Offset Surfaces. *ACM Trans. Graph.* 34, 4, Article 102 (jul 2015), 9 pages. <https://doi.org/10.1145/2766955>
- Przemyslaw Musalski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2016. Non-Linear Shape Optimization Using Local Subspace Projections. *ACM Trans. Graph.* 35, 4, Article 87 (jul 2016), 13 pages. <https://doi.org/10.1145/2897824.2925886>
- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (jul 2015), 12 pages. <https://doi.org/10.1145/2766937>
- Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Trans. Graph.* 32, 4, Article 81 (jul 2013), 10 pages. <https://doi.org/10.1145/2461912.2461957>
- L. Wang and E. Whiting. 2016. Buoyancy Optimization for Computational Fabrication. *Computer Graphics Forum* 35, 2 (2016), 49–58. <https://doi.org/10.1111/cgf.12810> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12810>
- Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. 2013. Cost-effective printing of 3D objects with skin-frame structures. *ACM Trans. Graph.* 32, 6, Article 177 (nov 2013), 10 pages. <https://doi.org/10.1145/2508363.2508382>
- Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. 2017. Infill optimization for additive manufacturing—approaching bone-like porous structures. *IEEE transactions on visualization and computer graphics* 24, 2 (2017), 1127–1140.
- Haiming Zhao, Chengkuan Hong, Juncong Lin, Xiaogang Jin, and Weiwei Xu. 2016. Make it swing: Fabricating personalized poly-poly toys. *Computer Aided Geometric Design* 43 (2016), 226–236. <https://doi.org/10.1016/j.cagd.2016.02.001> Geometric Modeling and Processing 2016.