Optimizing Contact-based Assemblies

DAVI COLLI TOZONI, New York University, USA
YUNFAN ZHOU, New York University, USA
DENIS ZORIN, New York University, USA

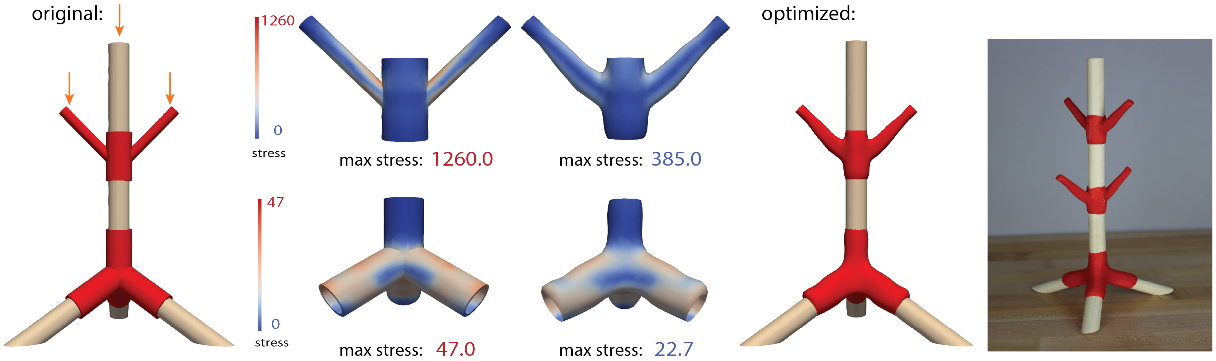

Fig. 1. Coat rack stress: stress concentrations on coat rack example. Left figure presents how load is applied. Middle figures show stresses on each PLA piece. Right figures show the optimized coat rack and 3d printed connectors assembled with wooden dowels.

Modern fabrication methods have greatly simplified manufacturing of complex free-form shapes at an affordable cost, and opened up new possibilities for improving functionality and customization through automatic optimization, shape optimization in particular. However, most existing shape optimization methods focus on single parts. In this work, we focus on supporting shape optimization for assemblies, more specifically, assemblies that are held together by contact and friction. Examples of which include furniture joints, construction set assemblies, certain types of prosthetic devices and many other. To enable this optimization, we present a framework supporting robust and accurate optimization of a number of important functionals, while enforcing constraints essential for assembly functionality: weight, stress, difficulty of putting the assembly together, and how reliably it stays together. Our framework is based on smoothed formulation of elasticity equations with contact, analytically derived shape derivatives, and robust remeshing to enable large changes of shape, and at the same time, maintain accuracy. We demonstrate the improvements it can achieve for a number of computational and experimental examples.

CCS Concepts: • **Computing methodologies → Mesh models**.

Additional Key Words and Phrases: Additive Fabrication, Deformable Objects, Shape Optimization

ACM Reference Format:
Davi Colli Tozoni, YunFan Zhou, and Denis Zorin. 2021. Optimizing Contact-based Assemblies. *ACM Trans. Graph.* 40, 6, Article 269 (December 2021), 19 pages. https://doi.org/10.1145/3478513.3480552

1 INTRODUCTION

Creating shapes optimized for a particular function is one of the main tasks in computer-aided design. Modern fabrication methods have greatly simplified the creation of complex free-form shapes at an affordable cost and opened up new possibilities for improving functionality and customization through automatic optimization.

Methods for constructing optimal shapes have enjoyed considerable attention in a variety of settings, such as large-scale architectural forms, engine parts, footwear, medical prosthetic devices and metamaterial structures. Notably, most of the work focuses on designing continuous structures, fabricated from one material, or a set of materials fused or glued together.

In this work, our focus is on supporting shape optimization for *assemblies*, and more specifically for assemblies that are held together by contact and friction, a setting which has received relatively little attention. At the same time, assemblies are ubiquitous, as most manufactured objects around us are assembled from separate parts, often made from different materials. For example, the steel legs of a table may be inserted into openings of a wooden or MDF top; a phone may have a snap-on plastic protector or cover; a prosthetic device is attached to the body with friction. While the specific mechanisms for holding objects together may vary broadly, they are all based on combining deformation with contact and, in many cases, friction.

In all these cases, contact plays a major role in the function and mechanical behavior of the assembled object. Shape optimization helps to achieve better performance or save on the costs of material for fabrication (for additive fabrication these two are closely related).

Authors' addresses: Davi Colli Tozoni, New York University, USA, davi.tozoni@nyu.edu; YunFan Zhou, New York University, USA, yz6675@nyu.edu; Denis Zorin, New York University, USA, dzorin@cs.nyu.edu.

0730-0301/2021/12-ART269 $15.00
https://doi.org/10.1145/3478513.3480552

Some of the important measures of performance of assemblies include total volume or weight, total deformation energy, maximal stress and more complex measures such as permeability.

In this paper, we present a formulation and a robust numerical method for computing optimized shapes in the presence of contact and friction. Compared to shape optimization tasks not involving contact, the problem is significantly more difficult to solve, as it involves complex inequality constraints required for handling collisions and friction. The resulting problem is non-smooth, and often hard to solve sufficiently accurately. Our overall approach is to use a smoothed version of the problem [Eck et al. 2005] amenable to standard optimization techniques on the one hand, and allowing us to approximate the desired solution as close as possible on the other hand.

Contributions. In summary, the contributions include

- A shape derivative-based formulation for optimization problems with contact and friction, building on [Maury et al. 2017];
- A novel shape-optimization framework based on FEM discretization of this formulation, capable of handling contact regions between two deformable objects as well as a deformable and a rigid object. It provides sufficiently accurate elastic deformation computations to support, e.g., max stress reduction.
- The frameworks supports conventional functionals (stress-based and volume) and new, contact-specific functionals (assembly and disassembly, parallel alignment) that ensure that connection strength is maintained and that at the same time the parts held together by contact can be assembled. The framework also supports optimization involving multiple load scenarios.
- We demonstrate a range of 2D and 3D examples of shape optimization, and qualitatively evaluate these examples using laser cutting and 3D printing to fabricate them and demonstrate the expected behavior.

2 RELATED WORK

There is a broad range of work on shape/topology optimization and related methods, but relatively few works were trying to solve problems with contact. We focus on related work on shape optimization with contact, and briefly mention other shape/topology optimization research that we rely on.

Shape optimization with contact. Some previous works have considered contact of a soft body with rigid surfaces, for example, [Beremlijski et al. 2014], [Haslinger et al. 1986] and [Herskovits et al. 2000]. While other some other works have studied the interaction of two or more bodies in contact, like recent works from [Maury et al. 2017], [Desmorat 2007] and [Stupkiewicz et al. 2010]. Most papers do not consider friction, and those which do often consider simplified (compared to the standard Coulomb) friction models as discussed in [Maury et al. 2017].

For contact models, there are two families of algorithms, Lagrangian and the penalization methods. The first type of methods adds Lagrange terms for the model constraints to the objective function and uses sub-gradient-based optimization to deal with the fact that the problem is non-smooth. Examples can be found in [Herskovits et al. 2000] and [Stupkiewicz et al. 2010] , where a bilevel approach and an augmented Lagrangian method are used. The second type of methods, penalization methods, use smooth approximations to the problem, which add terms to the variational formulation. Our method belongs to this category. For this type of method, the objective function is smooth, which considerably simplifies optimization. In our work, we choose to use penalization method following the mathematical model presented by Eck et al. [Eck et al. 2005]. In [Maury et al. 2017], a similar overall approach with level-set discretization. While level-set modeling has many advantages (e.g., allowing for easy topology changes) in our experience, it is not well-suited for handling important types of problems, in particular, those involving stress reduction, which is our focus in this work, as stress is harder to resolve precisely.

Stress minimization. Maximal stress minimization was considered in a number of papers; the formulation closest to ours is [Panetta et al. 2017], where *worst-case* optimization for periodic metamaterial structures is considered; it uses parametric periodic shapes, which are meshed for shape derivative computation, and does not consider contact. Earlier work on minimization of maximal stress is [Allaire et al. 2004], which applies topology optimization to design lightweight minimal-stress objects built from sequentially laminated composites. Another similar work is [Allaire and Jouve 2008], which applies the level-set topology optimization method to minimize the p-norm of stress. None of these works consider contact, and level-set methods (with Eulerian discretizations on a fixed grid) require impractically fine meshes for accurate optimization of high p norms of stress. Other works considering max stress include [Lian et al. 2017; Polajnar et al. 2017; Sonmez 2009; Van Miegroet and Duysinx 2007; Xia et al. 2012]. In the computer graphics community, [Stava et al. 2012] was one of the first works to introduce heuristic shape correction techniques that effectively result in stress reduction. [Zhao et al. 2016; Zhou et al. 2016] consider problems involving bounding von Misses tress.

Contact and friction modeling. The literature on contact is extensive, and we mention only few most closely related works here; for general theory see e.g., [Stewart 2001]. Typically, contact problems are viewed as constrained optimization problems, with per-element-pair constraints. In particular, contacts between deformable objects that we consider in this paper, are defined as constraints between surface primitives (triangles, edges and vertices); some examples include [Belytschko et al. 2000; Bridson et al. 2002; Harmon et al. 2009; Otaduy et al. 2009; Verschoor and Jalba 2019]. Penalty-based methods for handling these constraints are among the oldest, but were largely supplanted by constraint formulations and LCP (linear complemenarity)-based or SQP solution methods. [Harmon et al. 2009] developed a method for which progressively high penalties are applied as the distance decreases, growing arbitrarily large as the distances to the object decreases. A recent work [Geilinger et al. 2020] provides a differentiable method for solving dynamic problems with contact and Coulomb friction using a penalty based model combined with equality constraints for static friction, which can be used with gradient-based optimization. Also dealing with Coulomb Friction, [Ding and Schroeder 2020] proposes a penalty based solution that can be used coupling rigid to deformable bodies and material point method (MPM). Another recent method [Li et al. 2020] shares some aspects with the approach we use, specifically, smoothed version of constraints is used, in particular for friction. However, we use finite penalties, rather than infinite barriers for constraints, and add smoothing both to contact and friction formulations.

It is well-known (see, e.g., [Moreau 1973]) that friction introduces significant non-smoothness to solutions, and due to their dissipative nature, solutions of problems with friction cannot be obtained using energy minimization. A variety of iterative methods for solving resulting problem were developed [Alart and Curnier 1991; Daviet et al. 2011; Jean and Moreau 1992]. Nonsmooth solution methods were applied in recent work with some success [Macklin et al. 2019], but are very difficult to use in the optimization context. Smoothed version of friction were proposed both in simulation and optimization context in the works we have mentioned.

Contact-based Assemblies. It is also important to mention works designing contact-based assemblies using geometric techniques and not necessarily relying on elasticity simulations to achieve their goal. Works like [Panozzo et al. 2013; Vouga et al. 2012; Wang et al. 2019] consider stability for self-supporting surfaces. Other works [Sun and Zheng 2015; Ureta et al. 2016; Yao et al. 2017] consider the design of joints for more general objects, including furniture.

3 PROBLEM FORMULATION

3.1 Overview

We start with a high-level overview of the general problem we are solving. The input to our algorithm is a collection of 3D meshed objects, some of which may be in contact (the meshes will be updated in the process of optimization).

We solve the problem of the general form

$$min_{\Omega(p)} J(\bar{u}), \quad \text{s.t.}, F(u) = 0, B(u) \leqslant B^0 \tag{1}$$

The main components of this formulation include the following.

- The *unknowns* in the optimization are shape parameters p, in our case displacements of mesh vertices on the boundary, defining the domain $\Omega(p)$, and displacements u of the points of $\Omega(p)$ resulting from elastic deformation with contact. (We use u for the continuous solution of the elasticity problem, and $\bar{u}$ for the vector of displacements of vertices of a meshed $\Omega(p)$). Note that while u corresponds to the deformations of the shape $\Omega(p)$ in our simulation, displacements p (on $\partial\Omega(0)$) define how the *rest-state* object shape is changed by optimization.
- The *PDE constraint* $F(\bar{u}) = 0$ is a FEM-discretized elasticity equation for $\bar{u}$ on $\Omega(p)$with contact and friction, leading to a non-linear system of equations in $\bar{u}$.
- The *objectives* in our optimization are of the one of two forms below, as we have objectives with computation on the whole object domain (e.g., stress) or only on its boundary (e.g., disassembly objective)

$$J(u) = \int_{\Omega_u} e(u,x)dx, \text{ or } E(u) = \int_{\Gamma_u^o} e(u,x)ds \tag{2}$$

where both the domain of integration and the integrands depend on the unknowns, and e represents a pointwise measure (e.g., stress), computed as a function of FEM solution defined by the discrete vector $\bar{u}$. Each objective can be either an *optimization target* to minimize as in (1) or a part of an inequality constraint (upper bound on an objective) described below. As optimization targets, we consider L^p-norms of stress, yielding compliance for $p = 2$ and a close approximation of the maximum stress for large p and volume, of a part or the whole assembly. Other objectives are used primarily as constraints.
- The *inequality constraints*, critical for our formulation, are of the form $B(\bar{u}) \leqslant B^0$, where $B(\bar{u})$ is an integral of the type (2). L^p norm of stress (effectively allowing to bound maximal stress) and volume objectives can also be used as constraints, e.g., we can impose a bound on the maximum volume allowed. Other objectives used primarily in constraints are the *assembly constraint*, ensuring that two parts can be assembled together; *disassembly constraint*, ensuring that once assembled, the object does not fall apart; and parallel alignment constraint, which makes it more difficult to disassemble the object using any direction other than the disassembly one.

Shape derivatives. The key element of the optimization process is computing objective and constraint gradients with respect to $\bar{u}$, required for any first- or second-order optimization algorithm.

There are two main approaches to this problem: one can discretize the problem first, fixing the mesh for $\Omega(p)$ for all p, and a FEM basis. This converts the problem into a finite-dimensional nonlinear algebraic problem, and then compute gradients of the objectives and constraints with respect to p, the positions of boundary mesh vertices. The alternative is the "differentiate-first" approach. Specifically, for each objective and constraint $J(u)$, *before it is discretized*, we derive its *shape derivative*, a continuous analog of the gradient with respect to $\bar{u}$. The shape derivative is a functional $dJ[v]$, where v is a velocity field of the deformations of the domain $\Omega(p)$ (i.e., the displaced position of a point q is $q + tv$). $dJ[v]$ yields the rate of change of J as $t \longrightarrow 0$. dJ is defined by a function ρ on Ω, which can be obtained by solving a *PDE*, similar in structure to the elasticity equation (*the adjoint equation*). The advantage of the latter approach is that it naturally allows for remeshing and refinement (the adjoint equation solver is just a standard elasticity solver), which is essential for large changes in the object shape: any method used to solve large-deformation elasticity PDEs can be used to compute ρ, without fixing the discretization in advance.

The cost of each optimization step is approximately equal to solving two elasticity problems, the nonlinear elasticity problem, and the linear adjoint problem, to obtain the solution and shape derivative respectively.

In the rest of the section, we provide more specifics on the components of the formulation described above and their discretizations. The complete derivations are included in the supplementary material.

3.2 Notation

We use the following notation, also illustrated in Figure 2

- Ω: optimization domain (may consist of multiple objects)

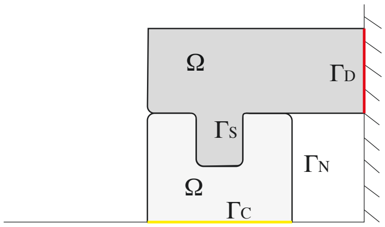

Fig. 2. Notation for boundary parts for a simple assembly: Γ_D is attached to the vertical wall (red), Γ_C can slide along the ground (yellow), Γ_N is where the load is applied, and Γ_S is the contact area of two parts. We view both parts as a single domain Ω consisting of detached parts.

- $\Gamma_D, \Gamma_N, \Gamma_C, \Gamma_S$: parts of the domain boundary where Dirichlet, Neumann, rigid-deformable (C) and deformable-deformable (S) contact boundary conditions are applied.
- superscripts +,– denote surface quantities on two sides of a contact surface between two deformable objects; $[\cdot]$ denotes the difference of two one-sided quantities.
- $u(x)$, $x \in \Omega$: displacements.
- $D[f]$ for a function f, its material derivative $\frac{\partial f}{\partial t} + u \cdot \nabla f$, where t is a deformation parameter.
- σ, $\varepsilon = \varepsilon(u) = \frac{1}{2}(\nabla u + \nabla^T u)$: stress and linearized elasticity strain.
- C: elasticity tensor.
- T: tractions on Γ_N, i.e., surface force density on the boundary.
- $1/\alpha$: penalty parameter for constraints;
- $\rho(x)$: solution of the adjoint PDE used to compute shape derivatives.
- ϕ_i, ψ_i: volumetric finite element basis functions at node i.
- λ_m: basis functions on the boundary.
- p: optimization degrees of freedom (boundary vertices of Ω)

3.3 Elastic deformations with contact

In this paper, we only consider static problems, so there is no time dependence in the equations we use.

The basic form of static equations of linearized elasticity (without contact or friction yet) is

$$\begin{aligned} \nabla \cdot \sigma &= 0 \quad \text{on } \Omega \\ u &= \hat{u} \quad \text{on } \Gamma_D \\ \sigma n &= g \quad \text{on } \Gamma_N \end{aligned} \tag{3}$$

The variational form of this problem, that we rely on to formulate the contact constraints in a computationally practical way, as well as for the finite element discretization, is given by the equation

$$\int_\Omega \epsilon(u) : C : \epsilon(w)\, dV - \int_{\Gamma_N} w \cdot g\, dS = 0 \tag{4}$$

satisfied for any w in an appropriate function space with $w = 0$ on Γ_D, and u satisfying the Dirichlet boundary conditions on Γ_D.

Contact constraints. We consider two forms of contact constraints, *rigid-deformable* (RD) and *deformable-deformable* (DD), defined on the parts of the boundary Γ_C and Γ_S respectively.

The former type of constraints involves a fixed boundary to which we refer as *obstacle*, on one side, which can be considered rigid, e.g., legs in contact with the floor.

$$\begin{aligned} u \cdot n &\leqslant 0 \quad \text{on } \Gamma_C \\ \sigma n \cdot n &\leqslant 0 \quad \text{on } \Gamma_C \\ (\sigma n \cdot n)(u \cdot n) &= 0 \quad \text{on } \Gamma_C \end{aligned} \tag{5}$$

The first equation on Γ_C says that the displacement should move the object points away from the obstacle; the second equation says that the normal force on this boundary should be 0 or point towards the other object; and finally, the last equation, the complementarity condition, ensures that if the force is nonzero, the displacement in the normal direction is zero, i.e., there is contact. The DD contact constraints are similar, however, because we have deformable material on both sides of the boundary, and the differences in stress need to be considered. Normals n^- and n^+ and displacements u^- and u^+ correspond to the parts on the opposite sides of contact, and $[u]$ is the jump $u^- - u^+$.

$$\begin{aligned} [u] \cdot n^- &\leqslant 0 \quad \text{on } \Gamma_S \\ [\sigma] n^- \cdot n^- &\leqslant 0 \quad \text{on } \Gamma_S \\ ([\sigma] n^- \cdot n^-)([u] \cdot n^-) &= 0 \quad \text{on } \Gamma_S \end{aligned} \tag{6}$$

Friction constraints. Similarly, there is a set of equations for friction:

$$\begin{aligned} &\|(\sigma n)_t\| \leqslant \mu |(\sigma n)_n| \text{ on } \Gamma_C \\ &\|(\sigma n)_t\| < \mu |(\sigma n)_n| \implies u_t = 0 \text{ on } \Gamma_C \\ &\|(\sigma n)_t\| = \mu |(\sigma n)_n| \implies u_t = -\lambda (\sigma n)_t \end{aligned} \tag{7}$$

where $\lambda > 0$, and μ is the friction coefficient; t and n refer to tangential and normal components of the force. The first inequality captures the main aspect of Coulomb static friction model (the force is bounded by μ times the normal force). The second equation states that if the force is below maximal, no displacement happens, and the third one that the displacement at maximal friction force is parallel to it and opposite in direction.

A modification (6) (replacing one-sided constraints with differences), applies to friction (please see the supplementary document), yielding *deformable-deformable* contact.

3.4 Variational form and constraint approximation

While the basic elasticity problem (4) is quite straightforward to solve and well-understood, contact constraints, especially friction constraints, result in numerous difficulties: (1) the system becomes highly nonlinear and non-convex; (2) the solutions may be non-smooth; (3) due to dissipative nature of friction, the problem cannot be cast as a (constrained) energy minimization problem. This makes even the direct solution of the problem difficult to make robust, and presents a particular challenge for shape optimization.

The key mathematical ideas for resolving these difficulties can be found in [Eck et al. 2005], further developed in [Maury et al. 2017]. (Recent work [Li et al. 2020] also follows a related approach for dynamic deformable contact).

The main elements of the approach include:

- Use *smooth* approximations of contact with friction (these are used to prove solution existence in [Eck et al. 2005], but

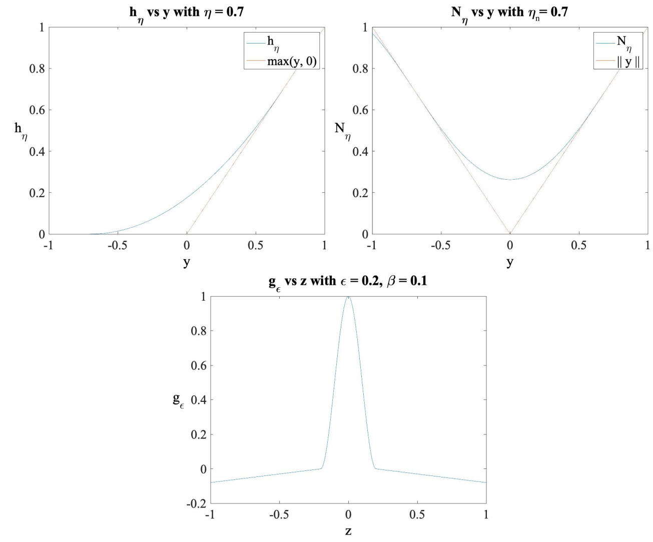

Fig. 3. Smoothing function N_η, h_η, and g_ϵ

are also valuable computationally). In particular, the norms in (7), that result in non-smoothness of the constraints are replaced by a smoothed norm that will be described in detail in this section.

- Replace the inequality constraints by suitable (also smooth) penalty functions, added to the variational formulation of the problem (4); a number of works (see [Maury et al. 2017] for discussion) show that the results converge to the true solution, as the penalty weight $1/\alpha \to \infty$. Like penalty approaches commonly used in graphics, using these methods has the benefit of replacing constrained optimization with unconstrained, and the downside of potential constraint violations which we discuss below.

We largely follow the smooth formulation of [Eck et al. 2005], which demonstrates that the solution to the smooth problem exists under typical assumptions, and with smoothing parameters approaching zero, converges to a solution of the original problem.

Smoothed contact functional. We introduce a smoothing function [Eck et al. 2005]

$$h_\eta(y) = \begin{cases} 0 & y \leqslant -\eta \\ \frac{1}{4\eta} y^2 + \frac{1}{2} y + \frac{\eta}{4} & -\eta \leqslant y \leqslant \eta \\ y & y \geqslant \eta \end{cases} \tag{8}$$

approximating $max(0, y)$ as $\eta \to 0$. This is the same function used in [Maury et al. 2017].

The contact equations (5) lead to the following objective

$$j'_{N,\alpha}(u, w) = \frac{1}{\alpha} \int_{\Gamma_C} h_\eta(u \cdot n)\ w \cdot n\, dS \tag{9}$$

As $\alpha \to 0$, the solutions obtained with this term added to (4) converge to solutions satisfying the constraint $u \cdot n \leqslant 0$. Similarly, we have

$$j'_{S,\alpha}(u, w) = \frac{1}{\alpha} \int_{\Gamma_S} h_\eta([u] \cdot n^-)\ [w] \cdot n^-\, dS \tag{10}$$

for DD contact.

Friction functionals. The Coulomb friction constraints (7) and the analogous constraint for DD friction, are expressed in terms of norms. For the smoothed functional, we use the following function to approximate $|y|$, as $\eta_n \to 0$:

$$N_\eta(y) = \begin{cases} \|y\| & \|y\| \geqslant \eta_n \\ -\frac{1}{8\eta^3}\|y\|^4 + \frac{3}{4\eta}\|y\|^2 + \frac{3}{8}\eta & \|y\| \leqslant \eta_n \end{cases} \tag{11}$$

Then friction constraints are captured by the following objectives [Eck et al. 2005]:

$$\begin{aligned} j'_{CF,\alpha}(u, w) &= \frac{\mu}{\alpha} \int_{\Gamma_C} h_\eta(u \cdot n)\ N'_\eta(u_t) \cdot w_t\ dS \\ j'_{SF,\alpha}(u, w) &= \frac{\mu}{\alpha} \int_{\Gamma_S} h_\eta([u] \cdot n^-)\ N'_\eta([u]_t) \cdot [w]_t\ dS \end{aligned} \tag{12}$$

Smoothed elasticity with contact and friction. Four objectives (9)-(12) are added to (3) to model contact with friction both for deformable-deformable and deformable-rigid contacts on parts of boundaries.

$$\begin{aligned} &\int_\Omega \epsilon(u) : C : \epsilon(w)\, dV - \int_{\Gamma_N} w \cdot g\, dS \\ &+ j'_{N,\alpha}(u, w) + j'_{S,\alpha}(u, w) + j'_{CF,\alpha}(u, w) + j'_{SF,\alpha}(u, w) = 0 \end{aligned} \tag{13}$$

for any w. The finite element discretization of these equations is standard and leads to a non-linear system of equations, because of the nonlinear functions. This system requires computing a Jacobian of the left-hand-side to solve efficiently. We refer to the supplementary document for the derivation of the Jacobian.

3.5 Optimization objectives

Next, we describe the optimization problem we solve, specifically, the set of objectives we use in the functional J as optimization targets or inequality constraints, and how their shape derivatives are computed.

In our model, we consider only the simplest type of assembly, namely, moving parts together in a specific direction y. This can be generalized to nonlinear trajectory settings (e.g., a screw motion trajectory) but this will make these constraints substantially more complex.

L^p stress and volume. These objectives are standard, and expressed as

$$J^p_\sigma(\Omega, u) = \left(\int_\Omega (\|\sigma(u)\|_F)^p dV\right)^{1/p}; \quad J_V(\Omega) = \left(\int_\Omega 1\, dV\right)^2 \tag{14}$$

The stress objective for large p approximates the non-smooth maximum-stress functional well. Following [Panetta et al. 2017], we consider the average stress at each element. Also, we use value $p = 20$ in most cases. We note that von Mises stress can be used just as easily. In addition, we have options for setting target stress and target volume, which are expressed as follows:

$$J^t_\sigma(\Omega, u) = \int_\Omega \varphi(\|\sigma[u]\|_F - S_t) dV; \quad J^t_V(\Omega) = \varphi\left(\int_\Omega dV - V_t\right) \tag{15}$$

where S_t and V_t represent the stress and volume targets and φ is the following function

$$\varphi(z) = \begin{cases} 0 & \text{if } z \leqslant 0 \\ z^2 & otherwise \end{cases} \tag{16}$$

We normalize our objective approximating maximal stress by the ratio between the load (in our boundary conditions) and the area of our shape's surface, which has units of stress. For the volume objective, we use $1/V_t^2$ as a normalization constant.

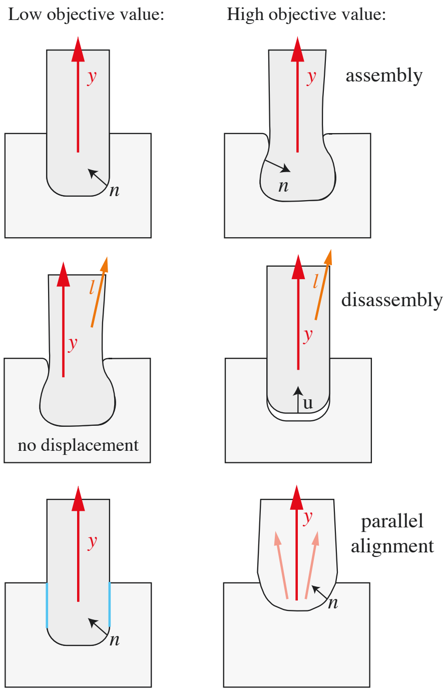

Fig. 4. Assembly, disassembly and parallel alignment objectives. Top: left shape is easy to assemble in $-y$ direction, while shape on the right cannot be assembled without effort (high objective value, due to normals pointing in the direction opposite to y). Middle: For the same disassembly direction y, and user-defined load l, the assembly on the right falls apart under the load; for the shape on the left, the parts cannot move apart in direction y under load l. Bottom: left shape has a large region parallel to disassembly direction (marked in blue), making it hard to disassemble in any direction other than y, while shape on the right allows for multiple easy disassembly directions (shown in light red).

We assume an disassembly direction y per connection, with assembly happening in opposite direction $-y$. We define two objectives, an assembly objective, which ensures that parts can be put together without much deformation in direction $-y$, and dissassembly objective, which penalizes parts moving apart along y, under the loads specified by the user.

Assembly objective. A fully physical treatment of assembly would require simulating the assembly process, leading to a large number of nonlinear solves for different time steps for a single gradient evaluation. Instead, we use a geometric heuristic with negligible cost to allow parts in contact to be assembled without a large deformation while moving a part along an a priori fixed direction in space. Assembly is ensured if the contact surface is a height field when we view y as a vertical direction. The angle between the direction of connection/disconnection and surface normal needs to be less than $\pi/2$. In our experiments, we always start with assemblable non-optimized parts, in order to facilitate the optimization.

This is not a hard constraint for deformable objects, as it can be violated by a small amount, to squeeze in a connector. We do not attempt to bound maximal possible deformation, but empirically it can be easily controlled by increasing the weight of this objective.

$$J^a(\Omega) = \frac{1}{|\Gamma_S|}\int_{\Gamma_S} \varphi(-n(x)\cdot y)ds \tag{17}$$

where y is the disassembly direction vector (and $-y$ the assembly direction), $n(x)$ is the normal at the contact position x, and $|\Gamma_S|$ is the area $A_S = \int_{\Gamma_S} 1ds$.

Disassembly objective. This objective ensures that under user-defined loads, or for a collection of different user-defined loads, the parts should not move away from the assembled positions. We formalize this by requiring that the optimized shape in equilibrium does not move in the disassembly direction, i.e., the forces holding contact points together are sufficient.

$$J^d(\Omega, u) = \frac{1}{|\Gamma_S|}\int_{\Gamma_S} \varphi(u(x)\cdot y \; - u_{tol})\, ds \tag{18}$$

A tolerance u_{tol} is added, allowing a small displacement in the assembly direction. While the assembly term is purely geometric, the disassembly objective relies on actual simulation responses and it bounds how much optimized parts can move. As long as it is finite, the assembly does not fall apart.

Although the last two objective terms may work against each other in the cases when the loads have a significant component along the disassembly direction, in other situations they work together to provide structures that are, at the same time, both assemblable and are not prone to accidental disassembly under user-defined load. See Figure 33 for an example.

Parallel alignment objective. Finally, this objective is a secondary heuristic that is not essential for solving the problems but empirically makes the structure more robust with respect to being disassembled by forces close to assembly direction, since it favors solutions with larger contact zones tangential to the assembly direction. When such zones are present, forces deviating from the disassembly direction have normal components resulting in deformations and friction keeping the parts from accidental disassembly. In the bottom of Figure 4, we can see how the shape without parallel alignment allows for easy disassembly in different directions (shown in red), without any reaction force (normal or friction), while the shape on the left allows for a single disassembly direction y.

This term is defined per continuous contact area Γ_S with respect to a disassembly direction y:

$$J^p(\Omega) = \varphi\left(W_{target} - \frac{1}{|\Gamma_S|}\int_{\Gamma_S} g_\epsilon(n(x)\cdot y)ds\right) \tag{19}$$

where W_{target} is the target percentage area in the connection which should be parallel to the disassembly direction, which can help increasing/decreasing reaction forces avoiding disassembly. The function g is parameterized by ϵ, defining the interval for which g is positive and β setting the linear decay of the function when $abs(z)$ is larger than ϵ. It has the following form:

$$g(z) = \begin{cases} \beta\,(z+\epsilon) & \text{if } z \leqslant -\epsilon \\ -az^3 + bz^2 + 1 & \text{if } z \leqslant 0 \\ az^3 + bz^2 + 1 & \text{if } z \leqslant \epsilon \\ -\beta\,(z-\epsilon) & \text{otherwise} \end{cases} \tag{20}$$

where ϵ is a smoothing constant and where a and b are defined as:

$$a = -\beta/\epsilon^2 + 2/\epsilon^3 \text{ and } b = -a\,\epsilon - 1/\epsilon^2$$

Summary. To summarize, our solver solves problems of the form (1), for which

$$J(u) = \sum_i k_i J_i, \tag{21}$$

where each of J_i is one of the objectives, enumerated above, combined with weights. The equality constraint $F(u) = 0$ is given by (13), the static elasticity equation with smoothed contact and friction.

The inequality constraints are of the form $J_m < J_m^{bnd}$ where J_m are some of the remaining objectives for which upper bounds are imposed (e.g., a bound on L^p norm of stress). These are imposed as soft constraints by adding $\sum_i \frac{1}{\epsilon_i}\phi(J_m^{bnd} - J_i)$ to $J(u)$.

4 SHAPE DERIVATIVES AND DISCRETIZATION

In this section, we describe our approach to computing the discretization of the problem and the gradient of the functional projected to the constraint space $F(u) = 0$ (i.e., space of displacements satisfying the elasticity equation with contact and friction).

4.1 Shape derivatives

For each objective, regardless of it is used as an optimization target, we need to compute the gradient with respect to p. As explained in Section 3.1, we use shape derivatives, which are computed using the solution of an adjoint equation. More specifically, the solution is used to construct a form dJ_d on the surface, which, when applied to changes δp of vertex positions on the boundary, produces the (linearized) change of the functional J:

$$dJ[\delta p] = \lim_{t \to 0} \frac{J(\Omega(p + t\delta p), u_t) - J(\Omega(p), u_0)}{t} \tag{22}$$

where u_t is the solution of the elasticity equations obtained on the domain with deformation $t\delta p$. The theory of shape derivatives is well-established, and we refer to [Bonnetier and Dapogny 2020].

Here, we only present a brief summary of the steps for computing these derivatives. A detailed derivation can be found in the supplementary material. Our formulation follows [Panetta et al. 2017], in that we consider volume, rather than surface integrals, which proved to be critical for derivative accuracy, at a moderate additional cost, compared to the elasticity solves involved.

Adjoint equation. The first step in computing dJ is to solve the *adjoint equation*, which in our case has the following form:

$$\int_\Omega \epsilon(\rho) : C : \epsilon(\psi)\,dV = \int_\Omega \tau : \epsilon(\psi)\,dS - (T'_C + T'_S + T'_{CF} + T'_{SF}), \tag{23}$$

for any ψ, where the terms T'_X, corresponding to the contact and friction smoothed penalties are of the form $\int_{\Gamma_X} F[u] \cdot \psi dS$, and ρ is the unknown. We observe that this is a *linear* elasticity equation which is relatively inexpensive to solve compared to the primal nonlinear elasticity equation. The quantity τ is defined as the sum of $2e'\sigma : C$ for all objectives of the form (2) where $e(s(\bar{u}), x)$ is a function on a stress measure, as our case. Then, e' means the partial derivative of e with respect to our stress measure.

Shape derivative. If the boundary deformation is defined by a set of basis functions λ_m , with the deformation expressed as $v = \sum_m \delta p_m \lambda_m$, where δp is the vector of changes of our variables, it is possible to express the shape derivative as a dot product

$$dJ\left[\sum_m \delta p_m \lambda_m\right] = S[u, \rho] \cdot \delta p \tag{24}$$

where $S[u, \rho]$ is a vector of the same length as p of vector valued functions depending on u and ρ. The expressions for these functions are included in the appendix, and derived in the supplementary document. We emphasize that no discretization, other than discretization of the deformation of the boundary was performed so far.

The vector $S[u, \rho]$ computed by numerical integration of FEM solutions of two elasticity problems for ρ and u is the gradient of the functional J with respect to p; this is what we use in the optimization process as discussed in Section 5.

4.2 Discretization

To make our discussion of the formulation complete, we summarize the discrete form of the problems we solve. The exact expressions are straightforward to derive but tedious (please see the supplementary material). We use quadratic Lagrangian elements on tetrahedras for discretization of (13) to obtain a system of the form

$$F(\bar{u}) = K\bar{u} - F_N + N_C(\bar{u}) + N_S(\bar{u}) + F_C(\bar{u}) + F_S\bar{u} = 0 \tag{25}$$

The first two terms come from the standard elasticity equation, and the rest correspond to the remaining terms in (13). The remaining terms, while simple, are not linear, so the system after discretization is a general algebraic system, and requires a non-linear solver. We also derive expressions for the Jacobians DN and DF, which are needed for efficient optimization.

In contrast, the adjoint PDE (23) is linear and has the form

$$(K + DN_C + DN_S + DF_C + DF_S)\,\bar{\rho} = D \tag{26}$$

i.e. involves exactly the Jacobians of the constraint functions. The right-hand side D is expressed in terms of τ. Formulas for the entries of these matrices are provided in the supplementary document.

Finally, once $\bar{u}$ and $\bar{\rho}$ are available, the coefficients of the shape derivative are computed from these values by integration, following formulas in the appendix, using numerical quadrature described in Section 5.

5 METHOD

Input. The input to our algorithm is a collection of tetrahedral meshes, with some parts of their surfaces in contact, and with boundary conditions applied to other parts of surfaces. Each object can be designated as deformable or rigid.

Our overall method is conceptually straightforward (Algorithm 1). The functions used in the pseudocode are:

- ElasticitySolve solves the nonlinear elasticity system (3), to obtain $\bar{u}$;
- AdjointSolve solves the adjoint system (26) to obtain ρ;
- DiscreteShapeDerivative, given $\bar{u}$ and ρ, computes the gradient $S[u,\rho]$ with respect to vertex positions on the boundary using (24).
- Converged is the outer iteration stopping criterion discussed below.

The inner loop works on a fixed connectivity for $\Omega(p)$, and is close to the standard BFGS algorithm: at each step, a descent direction is computed, and a line search is performed to determine our step size. There are three important differences: (1) we check for any inversions of tetrahedra and choose a step that maintains a bound on mesh element shape quality; (2) after each update of the boundary vertices, we call the SLIM smoothing algorithm [Rabinovich et al. 2017] on interior vertex positions p^{int} with boundary vertices p fixed, to move the interior vertices so that the quality of the mesh is improved; (3) with a valid mesh, we run simulation and if it doesn't converge in a maximum number of iterations, we reduce step by half in the line search.

If the step becomes too small, the inner loop is terminated, and the domain is remeshed in the outer loop.

A natural stopping criterion for the algorithm consists of three parts: (1) the objective reduction obtained in a step of (outer) iteration is below a threshold ϵ_r computed relative to the initial objective value; (2) the step size of the line search falls below a threshold (3) the maximal number of iterations *max_oi* is exceeded. Due to remeshing, the energy however may oscillate slightly, and for robust behavior we require that sufficient energy decrease does not happen over *m* steps. While many other options are possible (e.g., relative or absolute gradient norm threshold), we consider the rate of change in the objective to be most appropriate: our goal is to obtain a reduction in the objective, and a slow rate of reduction indicates that optimization will not improve the target by much more in a reasonable number of iterations; this happens either due to being close to a local minimum value or step size going to zero for geometric reasons, typically thin regions, resulting in distorted elements.

We discuss the choices for ϵ_r and m in Section 6, as well as examine convergence for specific test cases.

For remeshing, we use Triangle in 2D, and CGAL and fTetWild [Hu et al. 2020] for 3D tetrahedral meshing. More information on meshing is presented in the Appendix E.

Algorithm 1 contact optimization

```
function Gradient($p$)
    $u \leftarrow$ ElasticitySolve($\Omega(p)$)
    $\rho \leftarrow$ AdjointSolve($\Omega(p)$,$u$)
    $g \leftarrow$ DiscreteShapeDerivative($u$,$\rho$)
end function

function ShapeOptimization
    $p \leftarrow$ non-fixed boundary vertex positions
    repeat
        repeat
            $d \leftarrow$ BFGSdirection(Gradient, $p$)
            $\sigma, s, g \leftarrow$ LineSearch( Gradient,$d$)
            $p \leftarrow p + sd$
            SLIM($p^{int}$)
            $ii \leftarrow ii + 1$
        until $s < \delta$ or $ii = oi\_max\_ii$
        Remesh($\Omega(p)$)
        $oi = oi + 1$
    until $oi = max\_oi$ or not Converged
end function
```

Smoothness penalty. In addition to all optimization targets described in Section 3, we use a discrete regularization term $S_p(\Omega) = \sum_{v \in V} \|s(v)\|^p$ where

$$s(v) = \frac{\sum_{u \in N_v}(u - v)}{\sum_{u \in N_v}\|u - v\|}$$

and N_v is the neighborhood of v. The value of power p can be adjusted to obtain smoother surfaces at the cost of less optimal shapes; we use value 2 for most cases, increasing it to 4 for some objects. This term is scale-invariant and pushes the triangles/tetrahedra of the mesh toward equilateral.

Elasticity solver. In ElasticitySolve we use the standard Newton's method with line search to solve the nonlinear elasticity system with contact. We consider the simulation solved when residual is lower than a given tolerance. For 2D, we use 10^{-10} for the tolerance, while the value of 10^{-8} is used for our 3D examples. Moreover, for solving linear systems at each iteration of the Newton's method, we use CHOLMOD (for frictionless scenarios) and UMFPACK (when friction is present) in 2D. In 3D, we use MKL Pardiso library.

Adaptive Quadrature. An extremely important aspect of our implementation is the quadrature used to compute the integrals in the FEM system discretization, as well as in the shape derivative coefficient formulas. For problems with friction, precise computation of these integrals proved to be very important. At the same time, due to functions like h_η, the functions we integrate, while smooth, have higher derivative discontinuities. We use a combination of adaptive refinement on triangles and high-order (order equals to 10) Gaussian quadrature to integrate all functions to high precision. For more information, see Section D in the Appendix.

Convergence behavior. Figure 5 shows the objective as a function of iteration number for a 2D connector (Figure 9) and a 3D stool (Figure 24). For both cases, we optimized L_p norm of stress, while keeping volume below or equal the initial value. For the connector,

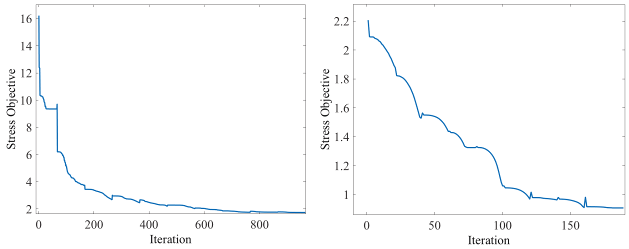

Fig. 5. Convergence plots for connector (Figure 9) on the left and stool (Figure 24) on the right side. The charts show stress objective vs iterations.

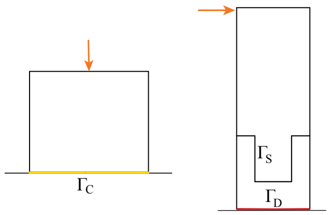

Fig. 6. Examples used for validation. The left: box in contact with a rigid surface. The right: joint in contact with a holder. We use the same notations for boundary parts as in Fig. 2. Starts from this Figure, all the orange arrows in the following figures illustrate the direction of load.

during the last 200 iterations, the energy decreased only by 0.87%, while for the stool, the minimum decreased by 0.27% in the last 30 iterations. We provide additional data in Section 6.

Observe that there are increases in energy at some of our iterations, corresponding to the remeshing of the current solution. However, after that the objective function quickly decreases, which motivates our formulation of the convergence criterion.

6 EVALUATION

Simulation Validation. As we use approximations to the standard physical models of contact and friction we evaluate the accuracy of these models. We use two examples. The first one is box in contact with a rigid surface. The second one is joint in contact with a holder as shown in Figure 6.

The results for both cases are largely the same. Thus, we will only show the results for box in contact with a rigid surface in this section and leave out the results for joint in contact with holder. We first change the value of α and η The reference value we use for comparison is obtained by simulation with $\alpha = \eta = 10^{-6}$. We set the smallest value of α to be 10^{-7} and scale it by the power of 2. The relative value of the difference in relative displacement $|u - u_{ref}|/|u_{ref}|$ is shown in Figure 7.

Then, we test the same scenarios but with friction at contact regions. First, we consider the dependence of accuracy on η_n, using $\eta_n = 10^{-4}$ as the reference value, as for friction we typically need more smoothing in the constraints. To test how friction coefficient affects simulation, we add in a small horizontal load that is 10% of the vertical load for the box example. Test with μ ranging from 0.1 to 0.16 and increase μ every time by 0.01 for the two examples. We compare the tangential displacement of the examples with respect to different μ to the tangential displacement of the examples with respect to $\mu = 0.1$ and take the ratio. We get the plots shown in Figure 8.

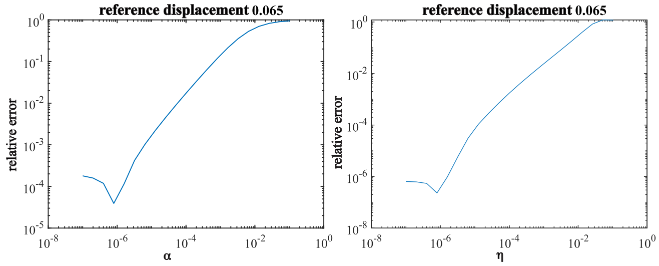

Fig. 7. Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing α. The right is relative displacement error with changing η

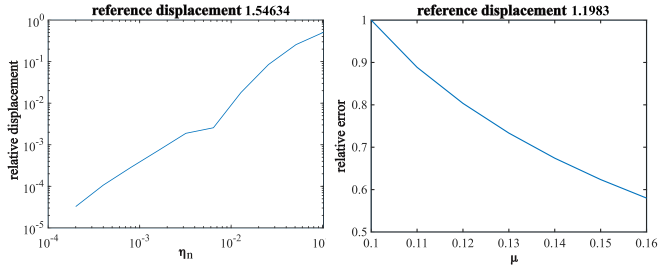

Fig. 8. Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing η_n. The right is relative displacement with changing μ

Effects of optimization targets and constraints. We use a simple two-dimensional connector example to demonstrate the effects of various optimization objectives and constraints (Figure 9) We compare stress distribution and maximal stress in different cases.

For most of the examples related to this scenario, we used a similar optimization configuration, running a maximum of 1000 iterations and remeshing at least every 100 iterations ($max\_ii = 100$ and $max\_oi = 10$).

The most radical difference is between minimizing the volume while bounding stress and minimizing maximal stress.

We also compared the scenario of minimizing maximal stress with and without a bound on volume (equal to the initial volume). In this comparison, stress results were actually very close to each other, with a slight advantage to the version with the constraint.

Multiple loads. Our framework allows optimizing for multiple separate acting forces, meaning that the energy related to stress will be a combination of the values from each separate scenario. An example is shown in Figure 10, where forces of the same intensity are applied to the right and to the left of the top piece. The stress results of the optimized shape are very close to each other and the final shapes present considerable symmetry, even though we are not enforcing any geometry symmetry through constraints.

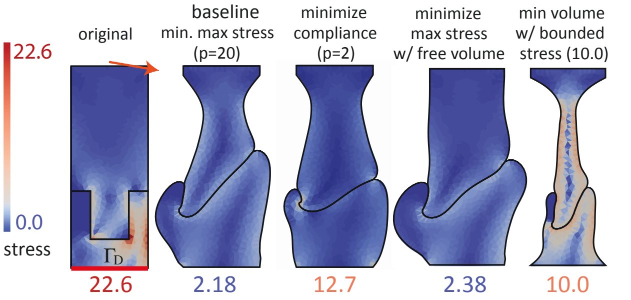

Fig. 9. Optimization results when using different combinations of objectives. For both baseline ($p = 20$) and compliance ($p = 2$) scenarios, we use a target volume equal to the original one.

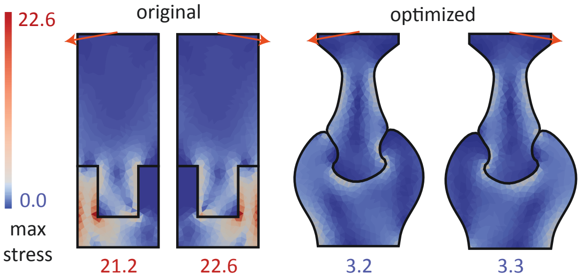

Fig. 10. Optimization considering multiple loads.

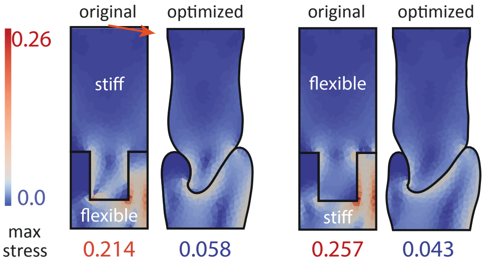

Fig. 11. Multimaterial optimization. The stiff material has Young's modulus equal to 100 and Poisson's ratio of 0.3, while the flexible one has Young's modulus equal to 1.0 and Poisson's ratio of 0.0.

Differences in material properties. Figure 11 shows how the results of optimization are affected by combining a highly flexible and a stiff material on our two different pieces. (In both cases the target is the baseline case of minimizing stress with no constraints). We observe that the stress reduction is similar, although shapes required for this are substantially different.

Role of friction. In the next comparison (see Figure 12), we observe that in the presence of friction ($\mu = 1.0$), we achieve a similar maximum stress value with the assembly not needing a more extreme protrusion to stay together, due to additional forces resulting from friction. This is demonstrated in Figure 12. The rightmost example shows the drastic effect of fixing only one part.

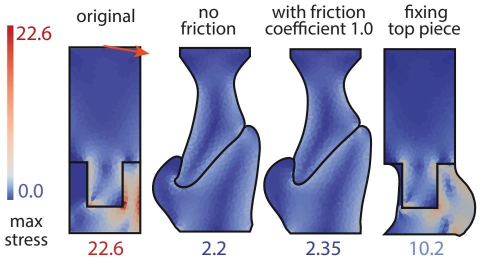

Fig. 12. Middle shapes: with and without friction; rightmost shape: the optimization of the top assembly is disabled.

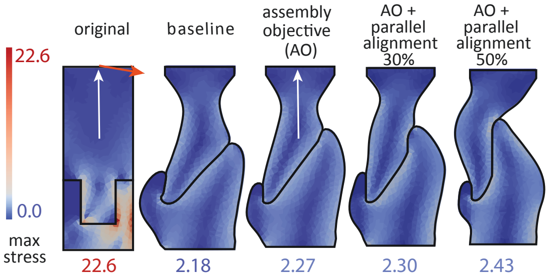

Fig. 13. Optimization results when using assembly and parallel alignment objectives, with two different strengths. The white arrow demonstrates the dissemble direction.

Assembly/disassembly and parallel alignment constraints. We show the effects of our connection-related constraints presented in Section 3.5.

Figure 13 shows the baseline optimization (minimizing maximal stress with a bounded volume) compared to different cases when our assembly constraint in all cases, and different target parallel alignments. Observe that, as expected, the contact area aligned with the disassembly direction increases.

All three examples are possible to assemble in vertical direction without deformation. Also, the parallel alignment objective guarantees that the desired proportion (30% and 50%) of the contact walls are parallel to the input direction.

Figure 14 shows the baseline case compared to one using a disassembly constraint with a very small tolerance. Compared to the the baseline optimization, two pieces do not detach when the disassembly constraint is applied. Another example of the importance of this constraint is shown in Figure 20.

Assembly/disassembly tradeoff. Finally, we also studied the effect of choosing different balance of assembly and disassembly objectives, in a setting when these counteract each other. Consider the case of Figure 15, where a force is pulling the top part of the connector up and we initially optimize the shapes using the same weight (100.0) for both terms and weight 1.0 for stress. Then, we reran our optimization using three lower weights for the assembly term. Figure 15 shows that, by reducing assembly's importance, almost no movement is observed in the last shape where assembly weight equals 0.1. When the disassembly term is dominant, the optimized shape has small protrusions on both sides that keep it firmly in

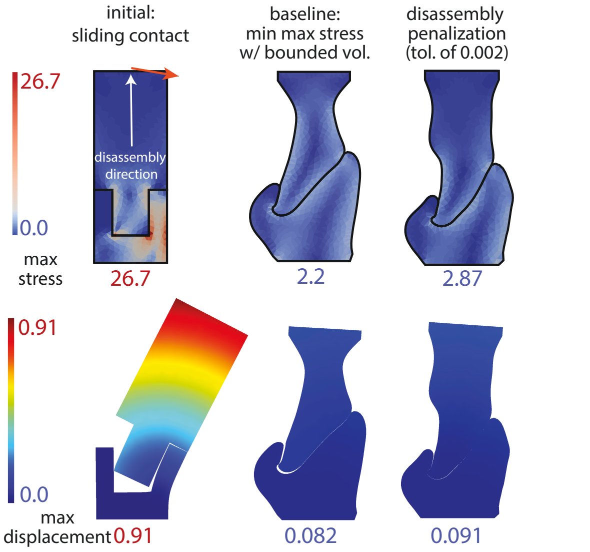

Fig. 14. Optimization results when using a disassembly energy term with low tolerance. As before, the white arrow demonstrates the dissemble direction.

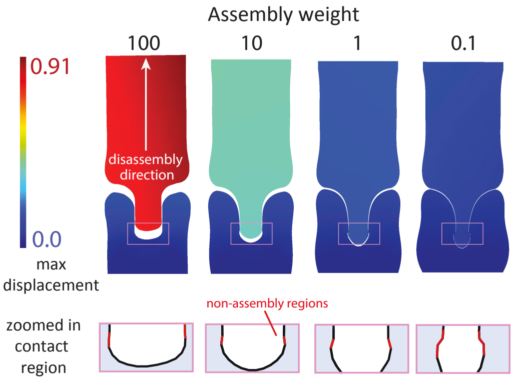

Fig. 15. Displacement results when optimizing our connector using varying assembly weights and a fixed disassembly weight.

Table 1. Final state of objective term values when running optimization with different assembly weights

	Assembly weight			
Objective	**100**	**10**	**1**	**0.1**
Assembly	0.02598	0.06968	0.35474	1.34750
Disassembly	0.01302	0.00275	0.00033	0.00003
Stress	2.53228	1.95349	1.35349	0.83647

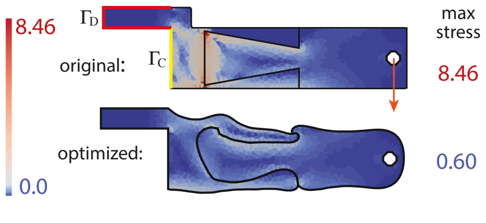

Fig. 16. Lever optimization

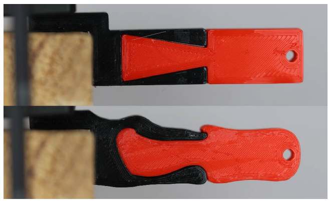

Fig. 17. Lever optimization fabrication

place, at the expense of greater effort required for (dis)assembling it along the (dis)assembly direction.

Table 1 presents a comparison of our main energies in this study. Note how assembly objective values increase when it's weight is reduced, while, at the same time, disassembly and stress values decrease considerably.

7 EXAMPLES

To validate our framework, we computed optimized shapes for a number of 2D and 3D realistic scenarios. In all illustrations in this section, parts of the object boundary with Dirichlet conditions are shown in red, and contact regions with external supports in yellow. We use $p = 20$ for stress optimization in all examples. Except for specific scenarios, in most of our examples we used the same objective function weights. A summary of our experiments in presented in Table 2, where we add information about the simulation, as well as objective functions used in each example, instance size and initial and final maximum values of stress.

7.1 2D examples

Lever. In this example shown in Figure 16, one part of the boundary of the black piece of the assembly is clamped to the table and in simulation assumed to satisfy the Dirichlet boundary condition; the part touching the vertical wall has an RD contact condition. Maximal stress is optimized, with no constraints imposed. As seen in Figure 16, we are able to reduce maximum stress more than 14 times compared to the initial shape. The optimized shape naturally evolves into an interlocking assembly that can support a far higher load without large deformation.(see the video in supplementary material). We fabricated resulting shape (the process is described in Appendix C, and performed a stress test on it, by loading it with increasing weight. Our setup is shown on Figure 17. The unoptimized lever breaks with a weight of $400g$, whereas the optimized lever can endure more than $6400g$, consistently with simulated results.

Bridge. Our second 2D example is a simple bridge model (Figure 18), with Dirichlet conditions on fixed rectangular parts, 3 optimizable parts, with boundaries of the supports partially fixed, and DD contact conditions between parts. Due to symmetry, we run the optimization on one half of the shape, resulting in 3 times stress reduction.

Table 2. Summary of experiments. In constraints column, "/" means that the weight is modified after each remeshing and varies between these values linearly. Column **Relative gradient norm** presents the ratio of the final gradient norm and the initial one from the first iteration. The last column (**Final L_p-norm reduction**) shows the reduction in the stress objective due to the last outer iteration, expressed as a percentage of the initial objective value. *For both bridge and coat rack (hooks), running 5 more outer iterations reduces stress to 1.80 and 346.51, relative gradient norm to 2.6e-2 and 2.81e-2, and final L_p-norm reduction to 0.95% and 0.00%, respectively. For fabrication, we used optimization results at outer iteration 5 for lever, hook, crane and at iteration 10 for bridge and coat rack.

Instance	Contact parameters $\alpha/\eta/\mu/\eta_n$	Iterations $oi\,/\,max\_oi$	Constraints (= weights)	Post vol. min.	BCs	Tets	Initial stress	Final stress	Relative gradient norm	Final Lp-norm reduction
2D examples										
Connector	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3$	no	Γ_D + DD + Γ_N	1250	22.6	2.18	2.38e-3	0.24%
Lever	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3$, $J^d = 10^2/10^3$	no	Γ_D + RD + DD + Γ_N	1447	8.46	0.49	2.88e-5	0.31%
Bridge*	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 200/2000$, $J^d = 10^2/10^3$	no	Γ_D + RD + DD + Γ_N	2073	7.33	2.21	1.55e-1	4.92%
Hook	$10^{-4}/10^{-4}/0.0/-$	9/10	$J_V^t = 10^2/10^3$	no	RD + Γ_N	1279	16.5	2.97	1.11e-2	0.00%
Hook (disassembly)	$10^{-5}/10^{-5}/0.0/-$	6/10	$J_V^t = 10^2/10^3$, $J^d = 10^2/10^3$	no	RD + Γ_N	1279	16.5	3.82	2.11e-2	0.74%
Crane	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3$, $J^d = 10^2/10^3$	no	Γ_D + DD + Γ_N	2087	8.47	1.09	7.81e-4	0.40%
3D multimaterial										
Stool	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	RD + DD + Γ_N	58264	2.68	1.12	3.23e-2	0.00%
Bench Twisted	$10^{-4}/10^{-4}/0.2/10^{-2}$	8/10	$J_V^t = 10^4/10^5$	no	RD + DD + Γ_N	12731	4.38	1.71	5.31e-2	0.00%
Bench Bended	$10^{-4}/10^{-4}/0.2/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	RD + DD + Γ_N	12362	2.82	1.99	1.23e-1	0.00%
Coat rack (base)	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	RD + DD + Γ_N	15464	47	22.7	2.28e-2	0.17%
Coat rack* (hooks)	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	Γ_D + DD + Γ_N	15726	1260	385	7.53e-2	2.16%
3D pipe-like										
Tent (baseline)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5$, $J^a = 10$	yes	RD + DD + Γ_N	15325	119	59.3	2.73e-2	0.24%
Tent (disassembly)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5$, $J^a = 10$ $J^d = 1.0$	no	RD + DD + Γ_N	15325	119	59.1	5.06e-2	0.84%
Tent (parallel)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5$, $J^a = 10$ $J^d = 1.0$, $J^p = 0.1$	no	RD + DD + Γ_N	15325	119	70.8	2.33e-2	0.08%
Truss	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5$, $J^a = 10$ $J^d = 1.0$, $J^p = 0.1$	yes	Γ_D + DD + Γ_N	29882	136	48.7	2.63e-2	0.15%
Dodecahedron	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5$, $J^a = 10$ $J^d = 1.0$, $J^p = 1.0$	yes	DD + Γ_N	23121	239	26.1	2.12e-2	0.00%

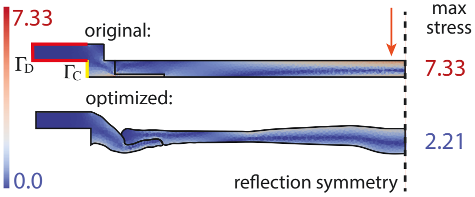

Fig. 18. Initial and optimized *bridge* model.

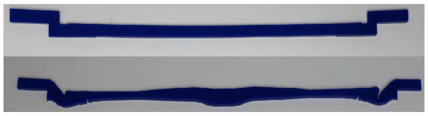

Fig. 19. Fabricated initial and optimized *bridge* shapes.

The left and right support parts of the bridge were clamped to the table and weights were attached to the middle of the bridge. The experiments show that unoptimized bridge breaks at the left or right assembly with a load of 1500g, whereas the optimized bridge can hold more than 5800g of weights. See photo of our bridge in Figure 19.

Hook. Figure 20, where we built an optimized *hook* that fits on the top of a door, with a single large contact area Γ_C, and *RD* boundary conditions on that part. In this scenario, we use multiple separate loads, one emulating, e.g., a coat hanging from the hook, and the other a force exerted when it is taken off the hook. In addition to multiple loads, this example also demonstrates the importance of disassembly energy: we make sure that accidental push from below does not result in the hook getting detached. The disassembly direction is shown in Figure 20. We perform stress minimization with a volume bound equal to the original volume, and compare the results with and without the disassembly constraint. Note that when the hook optimized without disassembly constraint is loaded, the left part lifts up, resulting in detachment from the support (Figure 20, middle) It remains stable once the disassembly constraint is enabled (Figure 20, right).

Figure 21 shows the experiments with 3 versions of 3d printed hook (unoptimized and optimized with and without disassembly

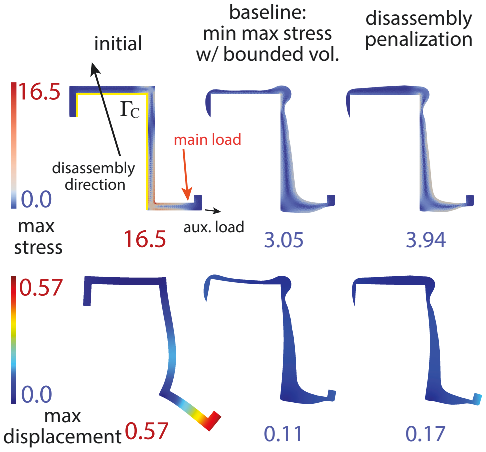

Fig. 20. Top: undeformed shapes of different versions of the *hook* model. bottom: deformations under load (displacements exaggerated).

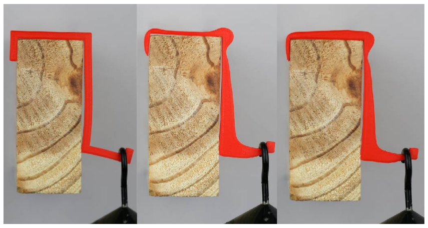

Fig. 21. 3D printed *Hook* loading (Please see supplementary video.)

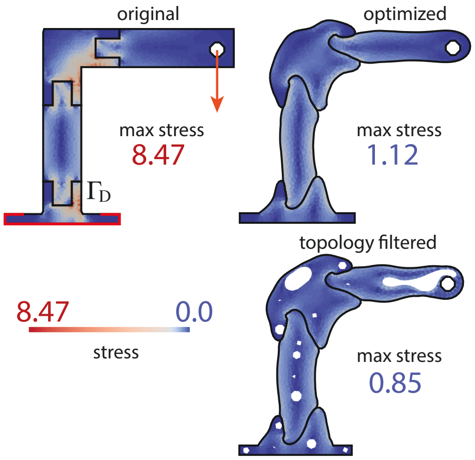

Fig. 22. *Tower crane* optimization

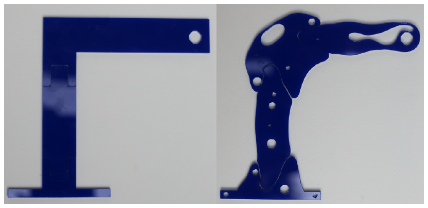

Fig. 23. *Tower crane* fabrication

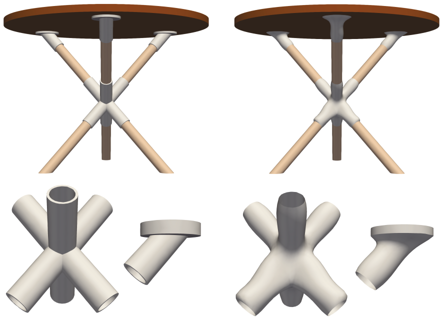

Fig. 24. Left: the initial *stool* model; Right: the result after optimizing the white (PLA) pieces.

constraint). The unoptimized hook breaks with 4500g load. The two optimized hooks can hold weights more than 19000g. Qualitatively, the behavior of loaded hooks matches the simulation.

Tower Crane. Our last 2D example is the tower shown in Figure 22, demonstrating an assembly consisting of four pieces with a Dirichlet condition on the bottom one, and DD contact conditions on the remaining parts.

In this example we demonstrate how our method can be combined with a simple ESO-like topology optimization technique (e.g., [Huang and Xie 2010]). to decrease the weight beyond what is possible with shape optimization only. We used a filtering technique to remove triangles with average stress lower than 10% of the maximum value. The filtered mesh was then again optimized, obtaining the result shown on the bottom-right. The laser-cut tower model is shown in Figure 23. In our experiment, the base of the tower was clamped to the table. Experiments show that the unoptimized tower breaks with 700g load, whereas the optimized tower can sustain more than 4600g load.

7.2 3D examples

In the next 3 examples, we apply our algorithm to 3D assemblies of parts made of different materials: wood ($E_{wood} = 10000, \nu_{wood} = 0.3$), MDF ($E_{MDF} = 4000, \nu_{MDF} = 0.25$) and PLA plastic ($E_{PLA} = 3500, \nu_{PLA} = 0.36$). Moreover, in these examples, only some pieces of the shape (those made of PLA, which can be 3D printed) were optimized.

Stool. Our first example was a *stool* assembly with a fixed MDF top, fixed wood legs and connectors made of PLA (Figure 24). In this example, we use volume constraints. The stress concentrations for optimized and noncan be seen on Figure 25.

The optimization reduces stress 2.5 times without changing the total volume. If 10% increase in the volume is allowed, the stress decreases to 0.88, and to 0.85 at 20%.

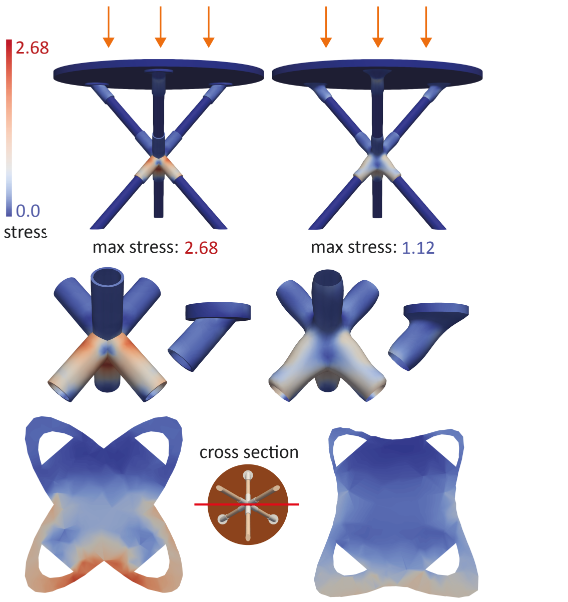

Fig. 25. Stool stress: stress concentrations on stool example. Top figures present how load is applied and general stresses (restricted to PLA). Middle figures show stresses on each PLA piece. Bottom figure shows cross section of region where stress is more concentrated.

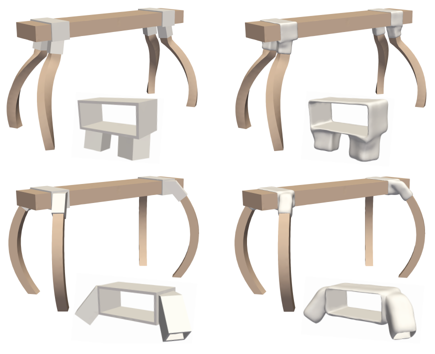

Fig. 26. *Bench* model with two different types of legs and the initial and optimized connectors.

Bench. Our second example is a *bench* with fixed parts (seat,legs) made of wood and optimizable PLA connectors (Figure 26). We consider two different leg shapes and show how connectors are optimized for each. We observe that, depending on the type of legs, we obtain a very different level of stress reduction: 30% in one case, and 2.5x times in the other (see Figure 27).

Coat rack. Our final multimaterial example is a *coat rack* made of wood and plastic (Figure 1). Differently from both previous examples, this object is modular, in the sense that you can always use a longer cylinder and add identical plastic parts to increase the height and the amount of hangers in your object. Here, we optimized our two different plastic parts for separate sets of loads, the leg connector and the hook.

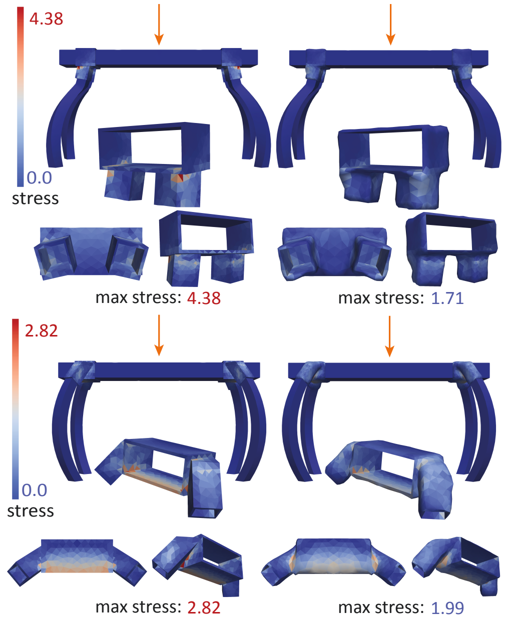

Fig. 27. Bench stress for twisted and bended set of legs

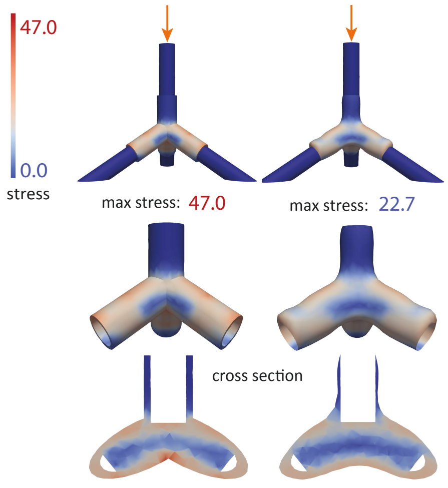

Fig. 28. *Coat rack* base stress distribution.

The leg connector is optimized with (fixed) legs in contact with the ground and the load applied at the top (Figure 28).

The hook attachment was optimized with loads applied to hooks. As shown in Figure 29, our framework was able to lower maximum stress more than 3 times.

We 3D printed the result using PLA (Figure 30).

For our last set of experiments, we implemented a framework for generating pipe assemblies from graphs, where each vertex of our graph becomes a sphere and each edge becomes a pipe. See an example on Figure 31.

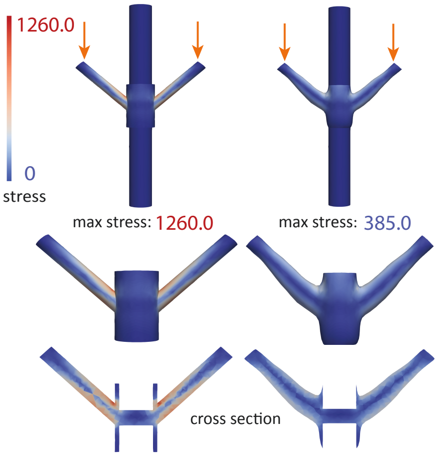

Fig. 29. *Coat rack* hook connector stress distribution.

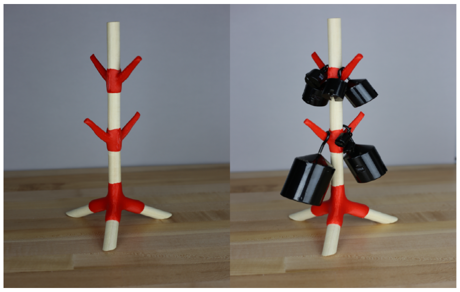

Fig. 30. Mounted coatrack

We used this tool to generate three different simple instances of our problem: a tent, a truss-like bridge and a dodecahedron. Below we detail more about each of these assemblies.

Tent. This object (Figure 31) is composed of 5 ball connectors and 8 pipes. A load is applied on the top of the structure pointing down and the bottom part of the object is in contact with the ground. We minimize stress, while keeping the same volume impose the assembly constraint. We were able to reduce stress by a factor of 2. We again ran our framework for reducing the volume of the shape, while keeping the stress bounded by the result of the previous optimization for stress (Figures 31 and 32). Then we add disassembly and parallel alignment objectives and the results are shown in Figure 33. Note that we optimize both the bars and connectors.

While the resulting structure can support this particular load well, it is likely to fall apart if any lateral load is applied, as resulting connectors are very shallow. Adding disassembly and parallel alignment constraints produces more resilient connectors (Figure 33). When using only disassembly constraint, the final stress (59.1) obtained was similar to the baseline case (59.3), while adding parallel alignment constraint to make at least 20% of the connection surface (in each connector) parallel to the disassembly direction increased stress to 70.8.

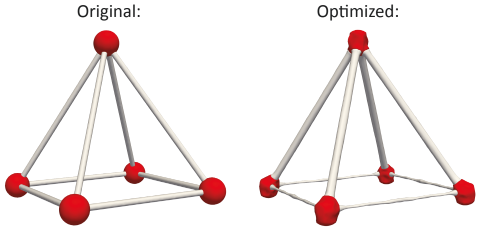

Fig. 31. Initial and optimized *tent* assemblies.

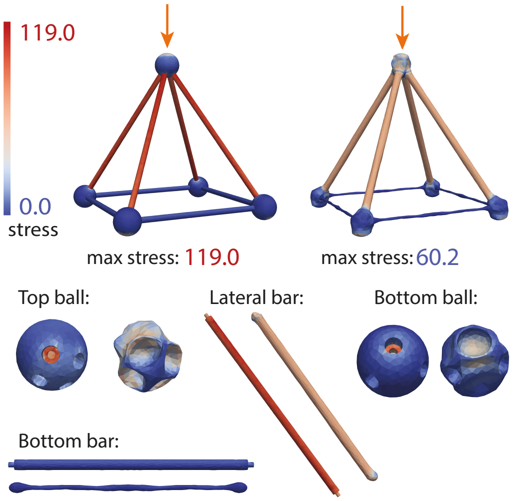

Fig. 32. Tent stress

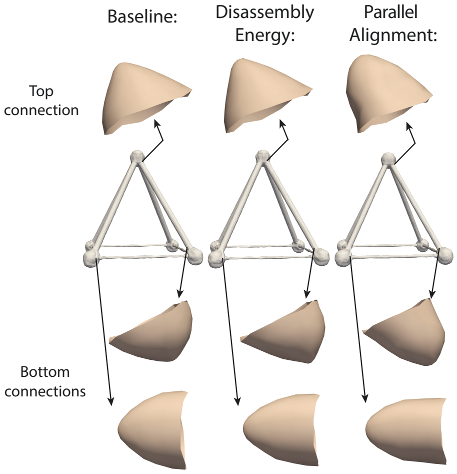

Fig. 33. Tent connections

Dodecahedron. This example is shown in Figure 34. For this example we use symmetric compression loads at the connectors at dodecahedron vertices. Using reflection symmetry on the three axes (x,y and z), we were able to run our optimization on an eighth of the

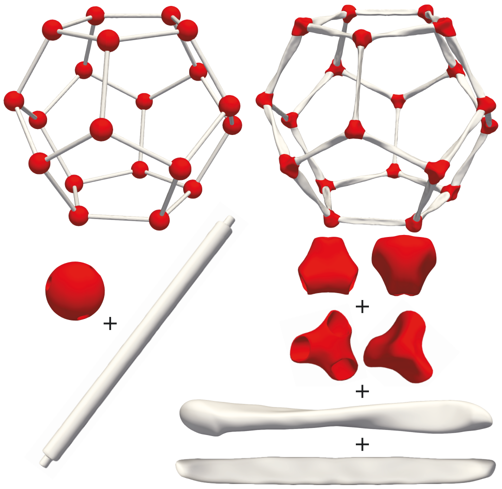

Fig. 34. Dodecahedron

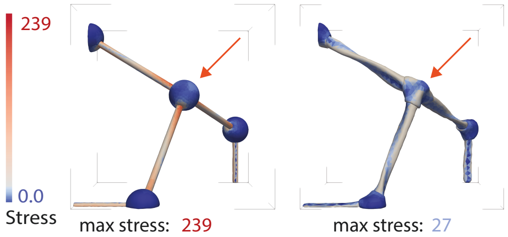

Fig. 35. Dodecahedron stress: simulation and optimization are executed on an octant of the original shape, using reflection symmetry on all three axes. A diagonal force to the center of the shape is used.

shape, as presented in Figure 35, where our boundary conditions are also shown.

We use a similar optimization setup as for *tent*: first, run stress minimization with a volume constraint, then minimize volume with a stress constraint, and including fixed assembly, disassembly and parallel alignment constraints. Notice in Figure 35 we were able to reduce stress by almost 10 times, while reducing volume in 12.6% compared to the initial dodecahedron. The emerging twisted shape for the bars are optimal for load support.

We also investigated the effect of optimizing only the connectors (balls) of the dodecahedron example. Again, we first minimize stress and then, in the second step, minimize the volume. We were able to reduce stress by around 28% to 172, and reduce the volume by 22%. The result can be seen in Figure 36.

Truss. The initial shape and the optimized result are shown in Figure 37. Similar to the dodecahedron, we optimized for strength (keeping initial volume) and then for volume (keeping stress close to its minimum level). We were able to reduce stress by more than half and volume by 20%. Notice that the optimized connectors are smaller than original ones while keeping deep connections. This result can be obtained thanks to the disassembly and parallel alignment energies.

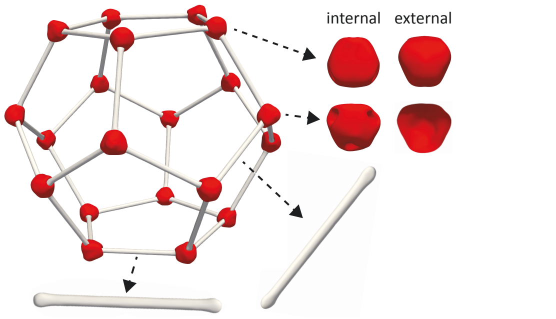

Fig. 36. Connectors optimization: result after optimizing only sphere connectors. We show our two different types of connector (showing two sides of each) and two different types of pipes, which were not optimized with exception of their tips.

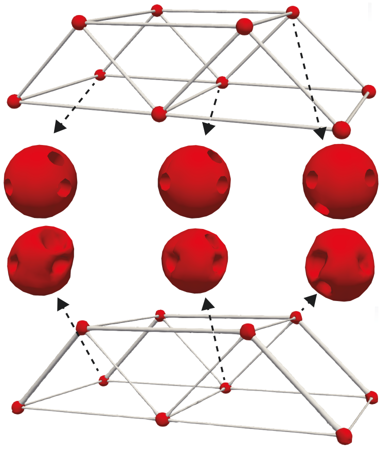

Fig. 37. Truss

Termination and convergence. To understand the convergence behavior of the method, we run our optimization for a fixed number of iterations which we aimed to set high enough for obtaining maximal possible improvement. Table 2 lists several indicators of convergence termination: number of actual iterations performed before stopping, relative gradient norm, and per-step objective reduction, expressed as a percentage of the initial objective value.

In the cases when the optimization reached the maximal number of steps (10 outer iterations for 2D and multi-material examples and 50 for 3D examples), the result is close to a local minimum, *or* the maximal number of iterations is insufficient. In most cases, relative gradient norm is 5% or significantly less of the original, indicating that the result is close to local minimum. In one case, (*Bridge*) additional iterations also decrease the relative norm below 5%.

In several cases (*hook* and two *bench* variations) the optimization stops for geometric reasons (in some cases, topology changes are needed to achieve an optimum); in this case, the optimization cannot reach a local minimum by additional iterations, although the gradient reduction is also high in most cases.

Examining how the objective changes with iteration suggests that values $m = 2$ and $\epsilon_r = 1\%$ are adequate for all examples (considering the criterion discussed in Section 5 for our stress objective), except truss-like structures, where values $m = 5$ and $\epsilon_r = 0.5\%$ are more suitable due to higher oscillations in the objective early in the optimization. We emphasize that many other convergence criteria can be used; determining an optimal one is not our focus.

8 CONCLUSIONS AND LIMITATIONS

We have described a framework for shape optimization with contact and friction, using a smoothed, penalty-based model, and applied it to optimization of assemblies in which parts are held together by contact and friction forces. We demonstrated that in our framework a number of functionals can be optimized reliably, producing significant improvements e.g., in stress concentrations or volume, while maintaining various types of constraints. We validated the results with computational experiments in two and three dimensions and with qualitative experiments using fabricated objects. The experiments confirmed significant improvements in strength. In addition, since we support nonlinear solves, our model can be easily extended to a variety of functionals and constraints. The code for this project will be open-sourced.

Limitations. Our algorithm may (1) not reach a minimum or (2) find a local minimum that does not improve the objective much. In the first case, the optimization stalls: no progress is possible for one of the following reasons: (a) nonlinear solve does not converge (e.g, if the system has no static equilibrium or bad-quality elements not allowing to achieve residual tolerance in max number of iterations); (b) inversions of elements lead to a severe line search step restriction, as mesh quality improvement is not guaranteed to succeed; (c) meshing with tolerance using fTetwild may produce volume meshes for pieces with non-manifold surfaces, while surface-preserving remeshing does not improve quality enough. In case (2), the algorithm may reach a local minimum without significant decrease of the objective value, while other minimal with lower values exist. Case (1) was the most observed in our experiments, occurring in our hook and bench examples due to narrowing regions during optimization; note that we still obtain a significant stress reduction.

In addition, we use linearized elasticity model that does not account well for large deformations. As the contact and friction terms in our equations are already non-linear and we use a nonlinear solver, extension to nonlinear elasticity is likely to be straightforward. While the functionals we have introduced in this work perform quite well for a number of tasks, there are restriction on the type of connections they can capture. E.g., snap connections are not handled easily with these constraints only. Expanding the range of objectives is an important direction for future work. The method can be considerably sped up but using parameterized geometric models, or model reduction, as well as more efficient solvers.

ACKNOWLEDGMENTS

Supported by the NSF grants DMS-1821334, OAC-1835712, OIA-1937043, CHS-1901091, and a gift from nTopology.

REFERENCES

Pierre Alart and Alain Curnier. 1991. A mixed formulation for frictional contact problems prone to Newton like solution methods. *CMAME* 92, 3 (Nov. 1991).

Grégoire Allaire and Francois Jouve. 2008. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements* 32, 11 (2008), 909 – 918. https://doi.org/10.1016/j.enganabound.2007.05.007 Shape and Topological Sensitivity Analysis: Theory and Applications.

G Allaire, F Jouve, and H Maillot. 2004. Topology optimization for minimum stress design with the homogenization method. *Structural and Multidisciplinary Optimization* 28, 2-3 (2004), 87–98.

Ted Belytschko, Wing Kam Liu, and Brian Moran. 2000. *Nonlinear Finite Elements for Continua and Structures.* John Wiley & Sons, Ltd.

P. Beremlijski, J. Haslinger, J. Outrata, and R. Pathó. 2014. Shape Optimization in Contact Problems with Coulomb Friction and a Solution-Dependent Friction Coefficient. *SIAM Journal on Control and Optimization* 52, 5 (Jan. 2014), 3371–3400. https://doi.org/10.1137/130948070

Éric Bonnetier and Charles Dapogny. 2020. An introduction to shape and topology optimization. (2020).

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. on Graph.* 21 (05 2002).

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Trans. on Graph.* 30 (12 2011).

B. Desmorat. 2007. Structural rigidity optimization with frictionless unilateral contact. *International Journal of Solids and Structures* 44, 3 (Feb. 2007), 1132–1144. https://doi.org/10.1016/j.ijsolstr.2006.06.010

Ounan Ding and Craig Schroeder. 2020. Penalty Force for Coupling Materials with Coulomb Friction. *IEEE Transactions on Visualization and Computer Graphics* 26, 7 (2020), 2443–2455. https://doi.org/10.1109/TVCG.2019.2891591

Christof Eck, Jiri Jarusek, Miroslav Krbec, Jiri Jarusek, and Miroslav Krbec. 2005. *Unilateral Contact Problems : Variational Methods and Existence Theorems.* CRC Press. https://doi.org/10.1201/9781420027365

Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: Analytically Differentiable Dynamics for Multi-Body Systems with Frictional Contact. *ACM Trans. Graph.* 39, 6, Article 190 (Nov. 2020), 15 pages. https://doi.org/10.1145/3414685.3417766

David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous contact mnumpageechanics. In *ACM Trans. on Graph. (TOG)*, Vol. 28. ACM.

Jaroslav Haslinger, Pekka Neittaanmaki, and Timo Tiihonen. 1986. Shape optimization in contact problems based on penalization of the state inequality. *Aplikace matematiky* 31, 1 (1986), 54–77. https://eudml.org/doc/15435

J. Herskovits, A. Leontiev, G. Dias, and G. Santos. 2000. Contact shape optimization: a bilevel programming approach. *Structural and Multidisciplinary Optimization* 20, 3 (Nov. 2000), 214–221. https://doi.org/10.1007/s001580050149

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 39, 4, Article 117 (July 2020), 18 pages. https://doi.org/10.1145/3386569.3392385

Xiaodong Huang and Yi-Min Xie. 2010. A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization* 41, 5 (2010), 671–683.

Michel Jean and Jean Jacques Moreau. 1992. Unilaterality and dry friction in the dynamics of rigid body collections. In *Proc. of Contact Mech. Int. Symp.*, Vol. 1.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 49–1.

Haojie Lian, Asger N. Christiansen, Daniel A. Tortorelli, Ole Sigmund, and Niels Aage. 2017. Combined shape and topology optimization for minimization of maximal von Mises stress. *Structural and Multidisciplinary Optimization* 55, 5 (2017), 1541–1557.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. (07 2019).

Aymeric Maury, GrÃ©goire Allaire, and FranÃ§ois Jouve. 2017. Shape optimisation with the level set method for contact problems in linearised elasticity. (Jan. 2017). https://hal.archives-ouvertes.fr/hal-01435325

Jean Jacques Moreau. 1973. On Unilateral Constraints, Friction and PLasticity. *New Variational Tech. in Math. Phys.* (1973).

Miguel Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross. 2009. Implicit Contact Handling for Deformable Objects. *Comp. Graph. Forum* 28 (04 2009).

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case stress relief for microstructures. *ACM Transactions on Graphics* 36, 4 (2017). https://doi.org/10.1145/3072959.3073649

Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. 2013. Designing Unreinforced Masonry Models. *ACM Trans. Graph.* 32, 4, Article 91 (July 2013), 12 pages.

https://doi.org/10.1145/2461912.2461958

Miha Polajnar, Franc Kosel, and Radovan Drazumeric. 2017. Structural optimization using global stress-deviation objective function via the level-set method. *Structural and Multidisciplinary Optimization* 55, 1 (2017), 91–104. https://doi.org/10.1007/s00158-016-1475-5

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1.

Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry Towards Geometric Engineering*, Ming C. Lin and Dinesh Manocha (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 203–222.

FO Sonmez. 2009. Optimal shape design of shoulder fillets for flat and round bars under various loadings. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 223, 8 (2009), 1741–1754.

Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomir Mvech. 2012. Stress Relief: Improving Structural Strength of 3D Printable Objects. *ACM Trans. Graph.* 31, 4, Article 48 (July 2012), 11 pages.

David E Stewart. 2001. Finite-dimensional contact mechanics. *Phil. Trans. R. Soc. Lond. A* 359 (2001).

Stanisław Stupkiewicz, Jakub Lengiewicz, and Joze Korelc. 2010. Sensitivity analysis for frictional contact problems in the augmented Lagrangian formulation. *Computer Methods in Applied Mechanics and Engineering* 199, 33 (July 2010), 2165–2176. https://doi.org/10.1016/j.cma.2010.03.021

Timothy Sun and Changxi Zheng. 2015. Computational Design of Twisty Joints and Puzzles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015)* 34, 4 (Aug. 2015). http://www.cs.columbia.edu/cg/twisty

Jane Tournois, Noura Faraj, Jean-Marc Thiery, and Tamy Boubekeur. 2021. Tetrahedral Remeshing. In *CGAL User and Reference Manual* (5.3 ed.). CGAL Editorial Board. https://doc.cgal.org/5.3/Manual/packages.html#PkgTetrahedralRemeshing

Francisca Gil Ureta, Chelsea Tymms, and Denis Zorin. 2016. Interactive Modeling of Mechanical Objects. *Computer Graphics Forum* 35, 5 (2016), 145–155. https://doi.org/10.1111/cgf.12971 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12971

Laurent Van Miegroet and Pierre Duysinx. 2007. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization* 33, 4-5 (2007), 425–438.

Mickeal Verschoor and Andrei C Jalba. 2019. Efficient and accurate collision response for elastically deformable models. *ACM Trans. on Graph. (TOG)* 38, 2 (2019).

Etienne Vouga, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. 2012. Design of Self-Supporting Surfaces. *ACM Trans. Graph.* 31, 4, Article 87 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185583

Ziqi Wang, Peng Song, Florin Isvoranu, and Mark Pauly. 2019. Design and Structural Optimization of Topological Interlocking Assemblies. *ACM Trans. Graph.* 38, 6, Article 193 (Nov. 2019), 13 pages. https://doi.org/10.1145/3355089.3356489

Qi Xia, Tielin Shi, Shiyuan Liu, and Michael Yu Wang. 2012. A level set solution to the stress-based structural shape and topology optimization. *Computers & Structures* 90 (2012), 55–64.

Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017. Interactive Design and Stability Analysis of Decorative Joinery for Furniture. *ACM Transactions on Graphics (TOG)* 36, 2, Article 20 (March 2017), 16 pages. https://doi.org/10.1145/3054740

Haiming Zhao, Weiwei Xu, Kun Zhou, Yin Yang, Xiaogang Jin, and Hongzhi Wu. 2016. Stress-Constrained Thickness Optimization for Shell Object Fabrication. In *Computer Graphics Forum*. Wiley Online Library.

Yahan Zhou, Evangelos Kalogerakis, Rui Wang, and Ian R Grosse. 2016. Direct shape optimization for strengthening 3D printable objects. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 333–342.

A FORMULAS FOR THE ADJOINT EQUATION TERMS.

$$
\begin{aligned}
T'_C &= \frac{1}{\alpha}\int_{\Gamma_C}\Big(h'_\eta(u\cdot n)\ (D[u]\cdot n)\Big)\ w\cdot n\, dS\\
T'_S &= \frac{1}{\alpha}\int_{\Gamma_S}\Big(h'_\eta([u]\cdot n^-)\ (D[[u]]\cdot n^-)\Big)\ w\cdot n\, dS\\
T'_{CF} &= \frac{\mu}{\alpha}\int_{\Gamma_C} h'_\eta(u\cdot n)(n\cdot D[u])\ N'_\eta(u_t)\cdot w +\\
&\quad + h_\eta(u\cdot n)\, N''_\eta(u_t)\Big(I-nn^T\Big)\, D[u]\cdot w \quad dS\\
T'_{SF} &= \frac{\mu}{\alpha}\int_{\Gamma_S}\Big(h'_\eta([u]\cdot n^-)\, n\cdot D[u-u^o]\Big)\ \Big(N'_\eta((u-u^o)_t)\cdot w\Big)\\
&\quad + h_\eta([u]\cdot n^-)\, N''_\eta((u-u^o)_t)\Big(I-nn^T\Big)\, D[u-u^o]\cdot w \quad dS
\end{aligned}
$$

B FORMULAS FOR THE DISCRETE SHAPE DERIVATIVE COEFFICIENTS $S_m[u,\rho]$

$S_m[u,\rho] = S^e_m + S^c_m$, where the first term is related to the elasticity equation and the second to contact and friction terms. In turn, $S^c_m = S_C + S_S + S_{CF} + S_{SF}$, the sum of terms for different types of contact forces. For these terms we have the following expressions, where we consider uniform surface load $g = \frac{G}{|\Gamma_N|}$:

$$
\begin{aligned}
S^e_m = &\left(\int_\Omega [e-\varepsilon(\rho):\sigma]\nabla\lambda_m + [\nabla\lambda_m\cdot(\sigma\rho_n + (\varepsilon(\rho):C-\tau)u_n)]\nabla\varphi_n dV\right)+\\
&\left(-\frac{1}{|\Gamma_N|}\left(\int_{\Gamma_N}\rho\cdot g\, d\Gamma_N\right)\left(\int_{\Gamma_N}\nabla\lambda_m\, d\Gamma_N\right)+\left(\int_{\Gamma_N}(\rho\cdot g)\ \nabla\lambda_m\ d\Gamma_N\right)\right)
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
S_C = -\frac{1}{\alpha}\int_{\Gamma_C} &h'_\eta(u\cdot n)\ (\rho\cdot n)\ (-(u\cdot\nabla\lambda_m)+(u\cdot n)(\nabla\lambda_m\cdot n))n\\
&+h_\eta(u\cdot n)\ (-(\rho\cdot\nabla\lambda_m)+(\rho\cdot n)(\nabla\lambda_m\cdot n))n\\
&+h_\eta(u\cdot n)\ (\rho\cdot n)\nabla\lambda_m\, dS
\end{aligned}
$$

$$
\begin{aligned}
S_S = -\frac{1}{\alpha}\int_{\Gamma_S} &h'_\eta([u]\cdot n^-)\ (\rho\cdot n)\ \Big[-([u]\cdot\nabla\lambda_m)+([u]\cdot n^-)(\nabla\lambda_m\cdot n^-)\Big]n^-\\
&+h_\eta([u]\cdot n^-)\ (-(\rho\cdot\nabla\lambda_m)+(\rho\cdot n)(\nabla\lambda_m\cdot n))n\\
&+h_\eta([u]\cdot n^-)\ (\rho\cdot n)\nabla\lambda_m\, dS
\end{aligned}
$$

$$
\begin{aligned}
S_{CF} = \frac{\mu}{\alpha}\int_{\Gamma_C} &-h'_\eta(u\cdot n)\rho\cdot N'(u_t)((u\cdot n)(\nabla\lambda_m\cdot n)-u\cdot\nabla\lambda_m)n\\
&+h(u\cdot n)(N''(u_t)\cdot\rho)\cdot(-u\cdot\nabla\lambda_m n+\\
&\ (u\cdot n)(\nabla\lambda_m\cdot n)n+(u\cdot n)(-\nabla\lambda_m+(\nabla\lambda_m\cdot n)n))n\\
&\ -h(u\cdot n)\rho\cdot N'(u_t)\nabla\lambda_m\, dS
\end{aligned}
$$

$$
\begin{aligned}
S_{SF} = \frac{\mu}{\alpha}\int_{\Gamma_S} &-h'_\eta([u]\cdot n^-)\rho\cdot N'([u]_t)(([u]\cdot n^-)(\nabla\lambda_m\cdot n^-)-[u]\cdot\nabla\lambda_m)n^-\\
&+h([u]\cdot n)(N''([u]_t)\cdot\rho)\cdot(-[u]\cdot\nabla\lambda_m n^-+\\
&\ ([u]\cdot n^-)(\nabla\lambda_m\cdot n^-)n^-+([u]\cdot n^-)(-\nabla\lambda_m+(\nabla\lambda_m\cdot n^-)n^-))n^-\\
&\ -h([u]\cdot n^-)\rho\cdot N'([u]_t)\nabla\lambda_m\, dS
\end{aligned}
$$

C FABRICATION PROCESSES

We have validated the performance of our system on several simple assemblies fabricated with 3D printing and laser-cutting. These are shown in Figures 17-18-20-22. For the lever and hook example as shown in Figures 17, 21, the fabrications are done with 3D printing (Ultimaker 3), using PLA material. For these experiments, we used Ultimaker Cura for slicing. We used fine printing setting, with 0.1mm layer height and infill density equal to 100%. The general print speed was also altered to 40mm/s and reduced to $12mm/s$ ($18mm/s$) when printing the inner (outer) walls of the object, in order to have a more precise printing of connections. In addition, to guarantee that objects connect, we offset the polygon shapes by 0.05mm, reducing them to account for the printing thickness.

For the bridge and tower examples (see Figures 18, 23), the experimented objects were obtained through laser-cutting 1/8" (3.175mm) acrylic boards, using Epilog Mini cutter. Notice that laser-cutting removes material, so we perturbed the models to account for that, offsetting the polygonal shapes by 0.05mm.

For our 3D examples, we also 3D printed our coat rack optimized model, shown in Figure 30. We used the fast setting on Cura software, with 0.2mm layer height, and reduced the speed to 50mm/s.

Even though we assembled with wood dowels, which can carry imprecision in the measures, the pieces still fit tightly.

D ADAPTIVE QUADRATURE

In different steps of our algorithm, during the simulation and also shape derivative computation, we need to perform the computation of integrals. For simpler cases using linear elasticity, this is not a challenge and we can simply use Gaussian quadrature, for example. However, when adding penalization due to normal contact and friction, the choice of how to perform these computations becomes very important.

As an example,consider the discretized normal contact force. Here, we assume we are using quadratic FEM elements. Our discretized contact function per contact element Γ_S^e is:

$$j'^{D}_{N,\alpha}(u) = \frac{1}{\alpha}\int_{\Gamma_S^e} \varphi^A(p(x))n_i\, h_\eta((u(p(x))\cdot n)\ dS$$

where α is a constant, n is constant for each element, $p(x)$ maps each point x into barycentric coordinates of the element, φ corresponds to the quadratic baricentric basis and u, the displacement, is also quadratic. h_η is the same as previously defined:

$$h_\eta(y) = \begin{cases} 0 & y \leqslant -\eta \\ \frac{1}{4\eta}y^2 + \frac{1}{2}y + \frac{\eta}{4} & -\eta \leqslant y \leqslant \eta \\ y & y \geqslant \eta \end{cases}$$

Notice that $u(p)\cdot n$ defines the region of the piecewise function. To find the boundary between each of these regions, we would need to solve $u(p)\cdot n = \pm\eta$. Notice that this corresponds to a quadratic 2D equation and the solution corresponds to a conic section: parabola, hyperbole, ellipse or circle. We have 2 equations, so our original domain is partitioned by a maximum of 2 conic sections. Since the two equations differ only by η, we expect that the regions boundary are going to have very similar shapes. See an example in Figure 38:

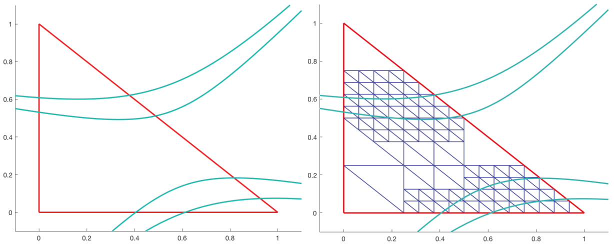

Fig. 38. Example of different regions in piecewise quadrature computation with $\eta = 10^{-2}$. In this case, we have two hyperbolas. So, in total, 5 different regions. On the right side, an example of how the triangles can be split.

To show the accuracy of the quadrature computation, consider the following example in 3D: $u_0 = [0.0, 0.1, -0.05]$, $u_1 = [0.1, 0.2; -0.03]$, $u_2 = [-0.1, 0.0, -0.02]$, $u_3 = [-0.05, 0.01, 0.02]$, $u_4 = [-0.02, -0.02, 0.01]$, $u_5 = [0.02, -0.03, 0.01]$, $n = \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right]$, $\alpha = 10^{-4}$, $\eta = 10^{-4}$ and nodes and coordinates $A = 0$ and $i = 0$. Here, u_j corresponds to the discrete displacement at each of our 5 nodes of our triangle element. Using Matlab's `integral2` function (with absolute tolerance set at 10^{-10}), we can compute $j'^{D}_{N,\alpha}$ to obtain -0.329516. Using (order 10) Gaussian quadrature, we obtain -0.438527. These errors can accumulate and generate problems in convergence for our simulation.

We implemented then an adaptive approach for computing integrals in our framework. First, we define the scheme to be used by the regions of our triangle. We evenly sample the triangle using m points, identifying the regions where these points are contained. In our case, we used $m = 10$. If not all m points are from the same region (easily verified by computing $u(p)\cdot n$ value), we split the triangle into 4 pieces of same area and repeat this process recursively, until a maximum depth is reached. See an example on the right side of Figure 38 In our case, we identified that depth equal to 5 guarantees good results. Inside each of the final triangles, we use (order 10) Gaussian quadrature to compute the integral. Using this method, for the same example as above, we obtained result -0.329560, which is accurate up to 4th digit after the decimal point when compared to the numerical result of the `integral2` function.

E MESHING

We use triangle meshes in 2D and tetrahedral meshes in 3D for our shapes.

For 2D, we define shape outlines using Adobe Illustrator, encoding boundary conditions using color; shared boundaries between parts are represented as single polylines. We then transform the geometry in the resulting SVG file into the input format for Triangle [Shewchuk 1996], removing the duplicate edges defining the contact area between different pieces. After obtaining the mesh for the interior, we split the triangle mesh into part meshes. At the end of each inner loop in Algorithm 1, we extract the polygonal boundary of the current solution and remesh it using Triangle again.

In 3D, the input to the meshing process is a collection of surface meshes of individual parts. We assume that in the contact zone, the distance between meshes is small, (in our specific examples we use boolean operations to produce surface for parts, with connectors obtained, e.g., by adding a cylinder to one part, and subtracting it for another, which produces close surfaces. After that, we generate the volumetric mesh using fTetWild [Hu et al. 2020] with surface meshes from the previous step as input. fTetWild has a threshold ϵ_m within which close parallel triangles of a surface mesh are merged; we set it to a sufficiently large value to obtain a single contact surface. For most cases, we used the default value of 1e-3 for ϵ_m, but increased the value to adjust to different gaps coming from the surface mesh generation.

As in 2D, we apply remeshing to keep a good mesh quality after a sequence of optimization steps. In 3D, remeshing is done in different ways: (a) preserving mesh surface, which is done with CGAL's package for tetrahedral remeshing [Tournois et al. 2021]; or (b) remeshing the boundary as well, by reapplying fTetWild. Choosing between this two options is also a parameter in our framework. We used option (a) for most of our scenarios, but for truss and dodecahedron example, boundary remeshing was used at every 5 remeshing operations. If remeshing the boundary with fTetWild fails, e.g., due to generation of pieces with non-manifold meshes, option (a) is used instead.