Soft Pneumatic Actuator Design using Differentiable Simulation

Arvi Gjoka
New York University / Courant
New York, USA
arvi.gjoka@nyu.edu

Espen Knoop
Disney Research
Zurich, Switzerland
espen.knoop@disneyresearch.com

Moritz Bächer
Disney Research
Zurich, Switzerland
moritz.baecher@disneyresearch.com

Denis Zorin
New York University / Courant
New York, USA
dzorin@cs.nyu.edu

Daniele Panozzo
New York University / Courant
New York, USA
panozzo@nyu.edu

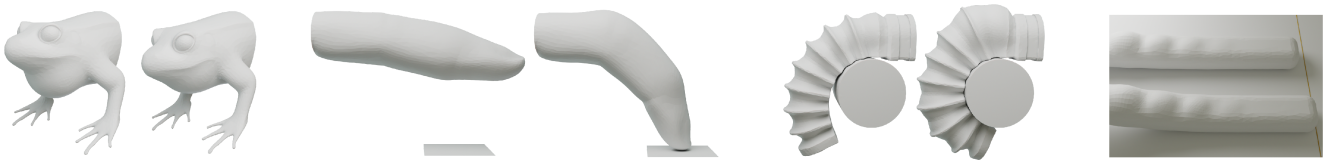


Figure 1: Shape optimization of a pressure cavity. We optimize the interior cavity of pressurized chambers to reach prescribed shapes (Frog, Finger, Gripper), displacements (Worm), and contact forces (Gripper). These results are then fabricated and tested experimentally to validate the efficacy of our pipeline.

ABSTRACT

We propose a computational design pipeline for pneumatically-actuated soft robots interacting with their environment through contact. We optimize the shape of the robot with a shape optimization approach, using a physically-accurate high-order finite element model for the forward simulation. Our approach enables fine-grained control over both deformation and contact forces by optimizing the shape of internal cavities, which we exploit to design pneumatically-actuated robots that can assume user-prescribed poses, or apply user-controlled forces. We demonstrate the efficacy of our method on two artistic and two functional examples.

CCS CONCEPTS

• **Computing methodologies** → **Modeling and simulation**; **Shape modeling**; **Simulation evaluation**.

KEYWORDS

Differentiable Simulation, Finite Element Method, Shape Optimization, Pneumatic Actuator, Soft Robotics

ACM Reference Format:
Arvi Gjoka, Espen Knoop, Moritz Bächer, Denis Zorin, and Daniele Panozzo. 2024. Soft Pneumatic Actuator Design using Differentiable Simulation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641519.3657467

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07…$15.00
https://doi.org/10.1145/3641519.3657467

1 INTRODUCTION

Pneumatic chambers embedded in soft materials is the predominant method to build soft robots that can reliably grasp fragile objects, locomote in challenging, obstacle-rich environments, or take on a desired shape if in- and deflated. Yet, it remains challenging to design soft pneumatic actuators that fulfill a set of artistic and functional requirements, especially if frictional contact is considered during the design phase.

In this work, we propose a differentiable simulation for shape optimization of soft robots that interact with the environment and satisfy a set of user-prescribed geometric and mechanical requirements. We extend the recently proposed differentiable incremental potential formulation to support pneumatic actuation and show that high-order finite element simulation with a Mooney-Rivlin material can faithfully capture large deformations of pneumatically-actuated soft robots.

We define a contact-aware objective that promotes high traction forces between a robot and an object, which is a common scenario in optimal actuator design for soft manipulation. We show that traction forces are too expensive to compute in optimizations, as they are only accurate if a dense mesh is used for simulation. We propose instead to use gradients of the contact potential as a proxy, as they are less sensitive to discretization.

The resulting inverse problem is computationally challenging to solve due to the sheer size of the problem (each iteration requires several forward simulations, which take up to tens of minutes) and the large number of local minima. We use a cascading optimization approach, together with a hierarchical shape parameterization with linear blend skinning subspaces, to tackle this optimization problem,

enabling large changes in shape at the first levels, which are then refined as the optimization proceeds.

Finally, we combine our algorithm with a simple, yet effective, modeling approach to use a CAD software as an interface to set up our optimization problem: both geometrical objects and optimization-specific selections (material, forces, functionals) are specified as surfaces of CAD objects. They are combined into a single FEM mesh using the TetWild algorithms [Hu et al. 2020] and optimized using our method.

We validate our approach on a set of simulated examples, and on three soft-robots that we fabricated with silicone and pneumatically actuate. Code and data for this paper can be found at https://github.com/arvigj/pneumatic-actuator-design.

2 RELATED WORK

We focus our review on related computational design techniques in computer graphics and computational explorations of pneumatic actuator designs in robotics.

Pneumatic Actuator Design. Early work in pneumatic actuator design focused on manual processes to demonstrate their versatility in building soft robots that are bio-inspired [Laschi et al. 2012], can locomote in difficult terrain [Shepherd et al. 2011], or manipulate fragile objects [Ilievski et al. 2011] with simple, imprecise control strategies, as surveyed by Rus and Tolley [2015]. Simulation-driven design explorations followed thereafter, digitizing the trial-and-error to improve their performance [Goury and Duriez 2018]. Optimization-driven design using a simulator in the inner loop [Bächer et al. 2021; Chen and Wang 2020] has received less attention, despite the difficulty of navigating the underlying unintuitive design space. Liu et al. [2014] use a level set method to optimize the topology of a gripper design with distributed compliance. Ma et al. [2017] describe a method to optimize pneumatic objects, not considering contact. Relying on topology optimization, Maestre et al. [2023] introduce a design optimization of a robotic skin that is pneumatically actuated. Our technique is also related to inflatable structure design [Skouras et al. 2014], but models soft robots whose designs vary in thickness, requiring solid instead of shell models.

However, we are unaware of a technique that is capable of making significant design changes in contact-rich scenarios, as addressed in our work.

Soft Object Design. Shape optimization techniques have received increasing attention in graphics, in particular in the metamaterial design context [Li et al. 2023a; Makatura et al. 2023; Panetta et al. 2017; Schumacher et al. 2015; Zhang et al. 2023]. Hafner et al. [2019] formulate a shape optimization approach that directly interfaces with CAD models. Focusing on flexible mechanism design, Maloisel et al. [2023] parameterize the shape of FE-discretized components using bounded biharmonic weights [Jacobson et al. 2011a]. We use a similar parameterization, but our technique supports significantly larger shape changes and considers frictional contact. The design of cable-actuated multimaterial soft objects [Skouras et al. 2013] or plush toys [Bern et al. 2017] has also been explored.

Pressurized membrane formulations. There is a large body of work on pressure formulations for membrane structures. A body of work [Bonet et al. 2000; Haßler and Schweizerhof 2008; Rumpel and Schweizerhof 2003; Schweizerhof and Ramm 1984] explores formulations for pressure forces acting on a cavity with fluid mixes. Niewiarowski et al. [2020] also use a similar formulation within an adjoint solver to optimize the shape of shell structures. In our work, we apply this formulation in the context of volumetric finite element simulation.

The closest work to ours in the graphics literature is by Skouras et al. [2012], which uses shape optimization on membranes to design balloons with a desired shape. We also rely on a shape optimization approach, but utilize a volumetric elastic formulation which can account and optimize for contact forces.

Differentiable Deformable Simulators. Numerous differentiable elastic body simulators have been developed for applications in optimal design of shapes [Baque et al. 2018; Beremlijski et al. 2014; de Vaucorbeil et al. 2019; Gavriil et al. 2020; Hafner et al. 2019; Hsu et al. 2022; Ly et al. 2018; Maury et al. 2017; Mitusch et al. 2019; Panetta et al. 2017; Stupkiewicz et al. 2010; Tozoni et al. 2020; Zhang et al. 2016], microstructures [Panetta et al. 2015; Schumacher et al. 2018; Tozoni et al. 2021], topology [Sharma and Maute 2018], actuators [Chen and Wang 2020; Hoshyari et al. 2019; Maloisel et al. 2021; Skouras et al. 2013], sensors [Tapia et al. 2020], material characterization [Bächer et al. 2021; Du et al. 2021; Hahn et al. 2019; Li et al. 2022; Liang et al. 2019; Schumacher et al. 2020], and robotic control [Bern et al. 2019, 2020; Chang et al. 2017; Geilinger et al. 2020; Heiden et al. 2021, 2020; Hoshyari et al. 2019; Hu et al. 2019a,b; Jatavallabhula et al. 2021; Luo et al. 2022; Qiao et al. 2020; Rojas et al. 2021; Xu et al. 2022a,b].

For our design application, the simulator must have the following properties: (1) high-accuracy simulation of deformable models, to be predictive of real-world behavior, (2) robust support for frictional contact with arbitrary geometry, as the interaction of the robot with the environment is mostly through contact forces, and (3) efficient support for shape derivatives of time-dependent simulations. To fulfill the last property, it should support efficient computation of shape derivatives which depend on thousands of parameters.

After a careful consideration of alternatives, we opted for using the open-source solver described in [Huang et al. 2023], adapting it to our purpose by adding support for pneumatic actuation and corresponding objective functionals. This solver is ideal in our setting as it uses a high-order finite element approach, supports contact using the recently proposed Incremental Potential Contact (IPC) [Li et al. 2023b], and has analytically-derived shape derivatives, implemented with the adjoint method. We refer to Huang et al. [2023] for more details on this approach and provide a self-contained summary in Sec. 3.

The main contributions over the differentiable simulator of Huang et al. [2023] are: (1) a multilevel optimization scheme that is independent of the initial design parameters, (2) a physically-accurate pressure formulation, corresponding actuation objectives, and experimental verification, and (3) physical validation of the overall system through the design of three soft robots.

3 PRELIMINARIES

We briefly overview the differentiable elastodynamic solver we build upon, to make our paper self-contained, and refer to [Huang et al. 2023; Li et al. 2020] for details.

Incremental Potential Contact Elastodynamic. We build upon the elastodynamic solver of [Li et al. 2020], where the displacement at the next time step u^{t+1} is computed as the solution of an *unconstrained* non-linear energy minimization:

$$u^{t+1} = \arg\min_u E(u, u^t, v^t) + B(x+u) + D(x+u) \tag{1}$$

where x represents the rest geometry, u the current displacements, and v^t current velocities. E is a time-stepping incremental potential (IP) [Kane et al. 2000], B is the barrier potential [Li et al. 2020], and D is the dissipative potential for friction [Li et al. 2020]. While we refer to [Li et al. 2020] for the complete formulation and details on contact and friction parameterization, we note that this work uses smooth potentials for collisions and guarantees that the geometry remains intersection-free throughout simulations.

Shape Optimization. Shape optimization computes a deformation of the rest pose, parameterized by a set of design variables q, which minimizes a user-prescribed functional J that depends on the outcome of the simulation,

$$\min_q J(u, x, q), \text{ such that } G(u, x, q) = 0, \tag{2}$$

where G is the gradient of the sum of the incremental potentials from Eq. 1.

To carry out this optimization, we write the corresponding shape derivative as

$$d_q J = \partial_q J + \partial_u J \partial_q u. \tag{3}$$

Since $G(u, x, q) = 0$, we can get an expression in $\partial_q u$ by applying the implicit function theorem,

$$\partial_q G + \partial_u G \partial_q u = 0, \tag{4}$$

which we can substitute into Eq. 3 and solve via the adjoint method

$$d_q J = \partial_q J - \partial_q G(\partial_u G \backslash \partial_u J) \tag{5}$$

We refer to [Huang et al. 2023] for a complete description. In our work, we add a new potential for pressure and propose a new pressure-specific objective to include in J.

4 METHOD

We hereafter provide a detailed description of our method, first introducing our functionals and then our cascaded shape optimization.

4.1 Pressure Boundary Condition

General Pressure Potential. The work done to inflate a closed chamber from V_0 to V_1 is given by

$$W = \int_{V_0}^{V_1} P(V) dV, \tag{6}$$

where, in general, the relationship between pressure P and volume V is unknown. However, we can write this dependence in closed form for special cases using the ideal gas law

$$PV = nRT, \tag{7}$$

where n is the amount of gas, R is the gas constant, and T is the temperature of the gas. From this equation, we can derive relationships between pressure and volume if we control for different parameters: keeping the temperature constant leads to an isothermal process $P = P_0(V_0/V)$ whereas requiring that no heat is lost leads to an adiabatic process $P = P_0(V_0/V)^\gamma$, where γ is a gas specific constant.

Isobaric Pressure Potential and IPC. The potential energy from Eq. 6 in a pressure chamber $\Omega(x+u)$ with volume $V(\Omega)$, whose shape is parameterized by $x+u$, at a fixed pressure P is

$$E_p(x, u) = P \cdot (V(\Omega(x+u)) - V(\Omega(x))). \tag{8}$$

This term is added to the IPC potential (Eq. 1). The derivation of the gradient and Hessian, which are needed by the IPC algorithm, is involved, as it depends on the deformed configuration. We follow the derivation in [Niewiarowski et al. 2020] that uses the divergence theorem to compute the volume $V(\Omega(x+u))$ as a surface integral. For reproducibility, we provide the derivation in our supplemental material.

Closed Cavity and Dirichlet BC. We note that this formulation assumes that $\partial\Omega$ is a closed surface. It can, however, be applied to open surfaces too as long as their boundary is held fixed by Dirichlet boundary conditions: the contribution to the energy of the missing part of the surface can be assumed constant and thus does not affect the gradient or Hessian computations. We will use this property in all our examples, as the cavities we are optimizing shape parameters for are *open* in our design to allow the connection to an external tube connected to a pump (Sec. 5).

Differentiable IPC. Computing the contribution of pressure to the shape derivative, $\frac{\partial G}{\partial q}$, is a straightforward process. Since the pressure term addition to the weak form is entirely a geometric quantity, $P\frac{\partial V(x')}{\partial x}$, and V is a function of $x+u$ locally, so that we can write

$$\begin{aligned} \frac{\partial \nabla E_p(x,u)[\phi_i]}{\partial x_j} &= \frac{\partial \nabla E_p(x,u)[\phi_i]}{\partial u_j} \\ \frac{\partial \nabla E_p(x,u)[\phi_i]}{\partial x_j} &= \Delta E_p(x,u)[\phi_i, \psi_j] \end{aligned} \tag{9}$$

where ϕ_i are the solution bases and ψ_j the geometry bases.

Pressure Control Derivatives. Using the boundary integral formulation, we can write $\nabla E_p(x, u)$ over a closed chamber as

$$\nabla E_p(x,u)[\phi_i] = P \oint_{\partial\Omega(x+u)} \phi_i \cdot \hat{n}(s)\, ds, \tag{10}$$

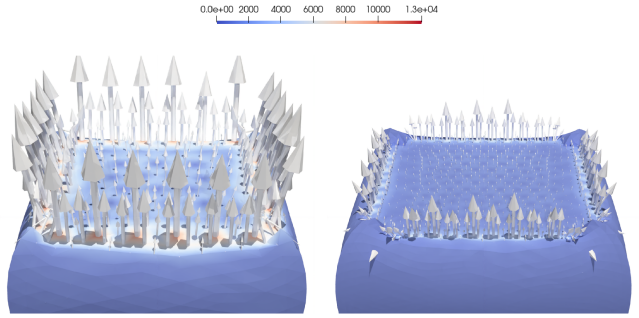
where P represents the pressure of the chamber. For the derivative with respect to the pressure value, $\frac{\partial G}{\partial P}$, we then have

$$\frac{\partial \nabla E_p(x,u)[\phi_i]}{\partial P} = \oint_{\partial\Omega(x+u)} \phi_i \cdot \hat{n}(s)\, ds. \tag{11}$$

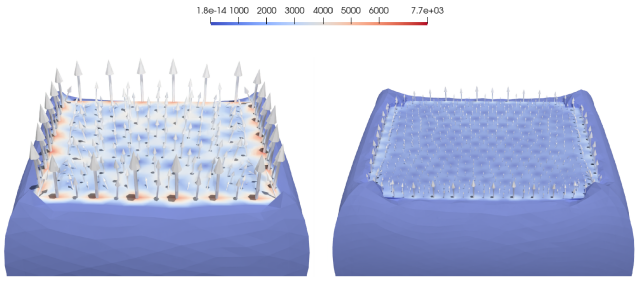
4.2 Contact Force Functional

Traction Force Functional. Traction is defined as

$$T = \hat{n} \cdot \sigma \tag{12}$$



Traction force



Gradient of potential

Figure 2: Simulation of the compression of a cube to 50% with Dirichlet boundary conditions on the top and bottom sides. We show two levels of refinement, 1101 **tetrahedra on the left,** 8808 **on the right. Surface traction forces integrated on triangles are unstable (top) leading to spurious forces mostly concentrated on the boundary loop. Under refinement, the effect is greatly reduced. In the bottom row, we show that the gradient of the potential is less susceptible to this effect and that the forces outside of the Dirichlet boundary are zero (to within solver tolerance).**

where $\hat{n}$ is the normal in the deformed configuration and σ is the stress tensor. Forces on surfaces can then be computed by integrating the traction over the area of interest. This is a natural quantity to optimize when designing robotic actuators for a variety of uses, such as when gripping an object.

Accuracy of Traction Forces. We show in Fig. 2 that directly using (12) is only accurate for fine meshes. For coarse meshes (especially if low order elements are used), the traction forces are not consistent with the force balance in finite element formulation: this quantity can be nonzero for regions with zero Neumann boundary conditions. We thus propose to compute the traction forces indirectly: instead of integrating traction forces on the object of interest, we integrate the contact forces that are acting on the same region. In the incremental potential formulation, these forces are the sum of the gradient of the contact and friction potentials. These are inherently discrete quantities, as the potentials are also discrete, and the computation is more stable (Fig. 2, bottom).

Energy Formulation. We formulate the contact objective as

$$J_c(x,u) = -\sum_{k\in C} \| \frac{\partial}{\partial x} b(d_k(x+u)) \|_2^2, \tag{13}$$

where C represents the set of all primitive pairs undergoing a collision, d_k the pointwise distance between elements in pair k, and b the barrier potential as a function of distance. By restricting C to pairs of surfaces of interest, we can localize this objective to specific parts of the scene, e.g. desired gripping surfaces. We measure only the magnitude of the forces as we are not interested in their direction.

IPC Local Support Limitation. For efficiency reasons, the contact potential E has a local support at a distance $\hat{d}$ from the surface. While this is a desirable property in forward simulation, as it reduces the computation cost and at the same time avoids contact forces to be added when objects are far from each other, it is undesirable for our objective: if the objective prescribes two surfaces to have contact forces, we would like the objective to be non-zero even if they are far away, so that this term can pull them close together.

Unfortunately, increasing $\hat{d}$ is not always possible, as $\hat{d}$ has to be smaller than the shortest edge in the mesh, otherwise IPC adds spurious self-contact forces. To avoid this issue, we ignore self-contact pairs in Eq. 13, as in our setting (inflation) self contact, while possible, is less likely. These changes apply only to the optimization objective and are completely separate from the contact formulation in the forward problem, which remains unchanged.

4.3 Minimal Thickness Functional

We introduce an objective penalizing small wall thicknesses. This is a typical fabrication requirement, as many 3D-printing and molding processes have a limit on how thin walls can be, which is particularly important in the case of pneumatically actuated soft robots. As pressure increases in a pneumatic chamber, assuming a uniform material, its thinner walls will deform the most, further reducing their thickness and thus making them more likely to fail as the pressure increases.

A general requirement for the objective is that it penalizes layer thicknesses smaller than a prescribed minimum. For this, we construct a term using the IPC distance, similarly to the contact potential,

$$J_l(x,u) = \sum_{k\in C} b(d_k(x+u), \hat{d}), \tag{14}$$

where C is the set of primitives for which we want to maintain a minimal separation. Similar to the contact force objective, we disable collisions between pairs of primitives on the same surface, as otherwise spurious forces would be added if the minimal distance is larger than the minimal edge length. This allows us to decouple the layer thickness parameter from the mesh discretization. In practice, we add this potential between surfaces that represent the inside and outside surface of a pneumatic chamber. Turning off collisions between primitives on the outside or inside surfaces is not harmful since we do not care about how the vertices are arranged on the surface from a layer thickness perspective. Since we are optimizing with a quasi-Newton method as opposed to Newton, we found convergence of a quadratic potential to be faster than with a log barrier in regions of complex geometry, such as within the folds of

the gripper in Fig. 8. We thus use a half-quadratic function for b:

$$b(d, \hat{d}) = \begin{cases} \kappa(d - \hat{d})^2 & d \leq \hat{d} \\ 0 & d > \hat{d}. \end{cases} \tag{15}$$

4.4 Shape Matching Functional

Another optimization objective we consider is matching a target geometry of an object containing a pneumatic chamber under inflation,

$$J_d(x, u) = \int_{\partial\Omega(x+u)} \|g(x+u)\|_2^2 ds, \tag{16}$$

where $\partial\Omega(x+u)$ represents the deformed surface of the mesh. $g(x)$ is a C^1 approximation of the distance function from x to the target surface defined as a tricubic interpolant on a regular grid [Lekien and Marsden 2005], interpolating the closest point function from each grid corner to the target surface. To avoid computing closest point queries on a densely sampled volume, we do a lazy evaluation of the interpolant by only constructing it in cells where it is needed. This objective allows to prescribe the desired shape of an object for a given pressure in its pneumatic chambers.

4.5 Cascaded Shape Optimization

Huang et al. [2023] propose to use L-BFGS [Nocedal and Wright 2006] to minimize the sum of objective functionals (Eq. 2), using the full finite element space coefficient as the optimization domain. We found that for pneumatic design, this approach tends to get stuck in undesirable local minima (Fig. 3). We propose an effective cascaded optimization which uses linear blend skinning subspaces to find more favourable solutions.

LBS Subspace. We parameterize a deformation of the rest pose by a set of rigid transformations, T_i, assigned to a set of n points $c_i \in P$ on the surface. The surface is deformed by blending these transformation using Linear Blend Skinning (LBS) for each vertex v,

$$v' = \sum_{c_i \in P} w_i(v) T_i v, \tag{17}$$

where the weights w_i are the Bounded Biharmonic Weights (BBW) computed using [Jacobson et al. 2011b], with the additional constraint that all weights must be zero on nodes fixed by Dirichlet boundary conditions and T_i are parameterized by three translational and three rotational degrees of freedom around the control points c_i.

Cascaded Optimization. The optimization algorithm is initialized with a user-provided number of vertices, n^0, a scaling factor, a, and proceeds in 4 steps for each iteration i:

(1) Uniformly sample n^i vertices on the surface.
(2) Compute the BBW for the corresponding LBS surface.
(3) Minimize Eq. 2, and apply the corresponding deformation to the rest pose.
(4) Increase the number of samples $n^{i+1} = n^i a$.

The algorithm terminates either when n^i is larger than the number of mesh vertices, or when it runs out of time. We used a max time of 7 days in our experiments.

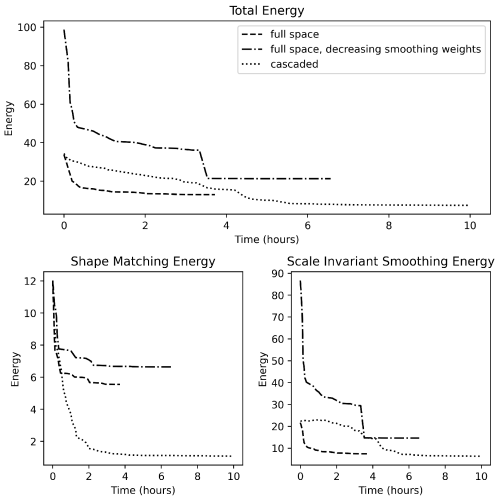


Figure 3: Convergence of cascaded and full space optimization. Full space optimization terminates early due to failure to find a valid step size (elements are nearly degenerate).

Vertex Sampling. We use furthest point sampling using geodesic distances to uniformly sample the surface and optimize their spacing using Lloyd's algorithm. We note that this procedure is performed using the optimized shape of the object so that the samples are uniformly distributed at each iteration of the algorithm.

Evaluation. We evaluate our cascaded optimization by comparing it with full space optimization on mesh vertices on one of our large examples. Fig. 3 shows the energy plotted over iterations and time for the optimization of a frog with an internal pressurized chamber outlined in Sec. 5. Firstly, we can see that cascaded optimization reaches a much smaller energy than full space optimization by their termination. This total energy is primarily made up of a shape matching energy J_d and a local smoothness energy (Scale-invariant Smoothing from Huang et al. [2023]), which are also plotted individually in Fig. 3. While the full vertex optimization decreases the smoothness energy at an early stage of the optimization, the cascaded optimization targets low frequency changes in the shape early on and only later decreases this high frequency smoothness energy. Consequently, it can explore the design space more efficiently than full vertex optimization and produce much better results. We attempt to match this behavior (first targeting low frequency changes) by having a higher smoothing weight that we then decrease throughout the optimization, but the results are similar to the case with a constant smoothing weight.

We give parameters for our optimizations in Tab. 1. The vast majority of time is spent in the forward simulation, with the gradient computation accounting for at most around 10% of the forward runtime. While some of the runtimes are high, the objectives being minimized can be highly non-convex which could lead to convergence difficulties for first-order methods like L-BFGS.

Table 1: Simulation and optimization parameters

Example	Number of tetrahedra	Runtime (hours)			n^0	a
		Total	Forward Simulation	Gradient Computation		
Frog	27391	10	8.0	0.2	1	2
Finger	16974	32	25.3	3.3	3	2
Gripper	49500	168	152.4	4.4	5	2
Worm	18588	134	113.4	8.6	1	2

4.6 From CAD to FE Models

To evaluate our method on complex geometries, we introduce a pipeline to go from an initial design in a CAD software to a final optimized shape, by using the CAD software as an interface. We use Solidworks to design our robot models, which are exported as surface meshes. We also use Solidworks to specify selections: a selection is a surface region of a Solidworks object, which is offset by a small value ϵ in both directions, resulting in a closed surface mesh containing the region of interest (we use ϵ equal to 10^{-3} times the diagonal of the bounding box).

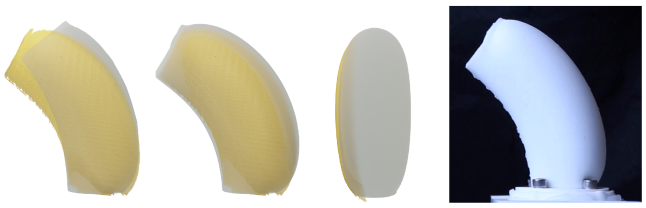
To set up the optimization, we use the volume mesher fTetWild [Hu et al. 2020] to create a volumetric mesh from all the surface meshes exported from Solidworks (excluding the selections). We set the fTetWild envelope size to 0.9ϵ to ensure that the parts of the input corresponding to the surface selections does not leave the offset. Then, we extract the boundary of the generated volumetric mesh and assign selection ids to boundaries enclosed by the user specified offsets. If a triangle is contained in two offsets, we always assign it to the first one. These selections are then used to assign both boundary conditions and objectives.

We use this pipeline for all our examples, which reduces the cost of iterating on the designs, as it hides most of the complexities in creating high quality meshes and setting up the finite element simulation. We note that while our input geometries are CAD files (i.e., NURBS models), our output is a dense triangle mesh corresponding to the boundary of the volumetric mesh used for simulation.

5 RESULTS

To demonstrate the utility of our approach, we validate our overall modeling with a pneumatic chamber inflation example, and optimize the design of two artistic examples to imitate the respiratory animation of a frog and the natural bending of a human finger. Targeting applications in robotics, we also optimize the function of a soft robotic gripper design and a simple locomoting soft robot.

Pneumatic Chamber Inflation. To test the pressure chamber simulation, we design a simple model consisting of nested elliptic cylinders. By making them not concentric, wall thicknesses vary continuously over the model and the model bends when the interior cavity is pressurized. This experiment is used to test the accuracy of the pressure formulation after we calibrate the silicone material for fabrication (see supplemental material). We show the results of the simulation along with the physical experiment in Fig. 4. After simulation with both P1 and P2 elements, we see that a low order solution basis exhibits a significant amount of error as compared



(a) P1 inflation (b) P2 inflation (c) P2 profile (d) Experiment

Figure 4: Inflation of a cylindrical chamber.

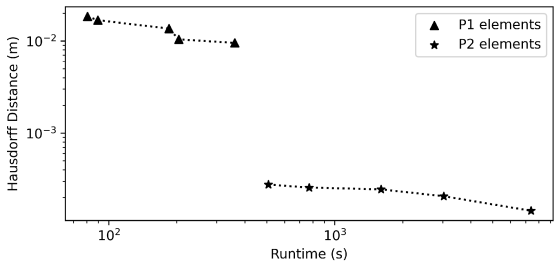


Figure 5: Refinement study on P1 and P2 elements. Hausdorff distance between a reference solution computed using P3 elements and the solution computed using linear (P1) and quadratic (P2) elements for different mesh resolutions for the example in Figure 4.

to quadratic elements. We conclude that a quadratic solution basis is necessary for simulations of pneumatically-actuated soft objects due to the large amount of locking observed with linear solution bases. With quadratic elements, the simulated results match closely the experimental results both visually and after doing surface reconstruction (orange overlay) and overlaying to the simulation result.

We also conduct a refinement study on this model by simulating a reference dense mesh using a cubic (P3) solution basis and computing errors of P1 and P2 solutions at different levels of refinement. This is shown in Fig. 5, where the coarsest P2 solution is more than an order of magnitude more accurate than P1 solutions for a similar running time. One could use an even coarser mesh to reduce runtime while keeping the accuracy lower than P1, however it gets increasingly difficult to approximate curved geometry with low resolution linear tetrahedral meshes.

For high-order simulations with contact, we use the method proposed by [Ferguson et al. 2023], whereby a finer linear approximation of the curved surface is used for contact.

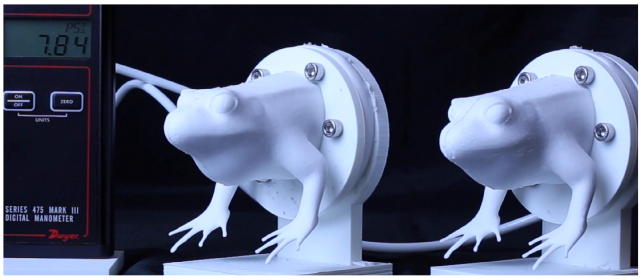
Closed Chamber Compression. We validate our pressure implementation by fabricating a silicone object with a closed cavity and compressing it using a column testing machine. We isolate the effects of pressure by producing the same cavity with a hole through which air can escape and no pressure builds up. We then compare the results with simulation, with our pressure implementation and without, respectively. Fig. 10 demonstrates the close match between

(a) Baseline chamber

(b) Optimized chamber

(c) Simulated baseline inflation

(d) Simulated optimized inflation

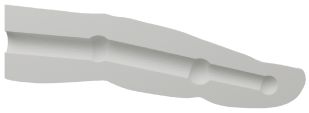


(e) Experimental inflation

Figure 6: Shape optimization of the interior cavity of a frog to minimize movement of the head and arms while puffing the neck.

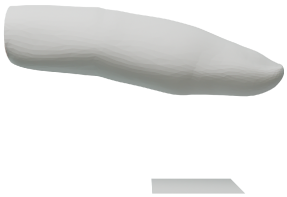
simulation and experiment quantitatively, whereas quantitative comparisons are given in supplemental material.

Frog. We design an experiment to test the ability of our optimization algorithm to match an artistic target in a quasistatic simulation. Starting with a model of a frog, we design a pneumatic chamber by naively adding a cavity inside. Upon inflation to 55 kPa (8.0 psi), we note the displacement of the surface and mark areas on the surface where the displacement is desirable. The goal for this example is to get the neck of the frog to distend while the rest of the frog remains unchanged under inflation. We then optimize this scene and fabricate the starting and optimized geometries. The objectives used for optimization are: (1) target matching to the artistic target, (2) an objective biasing against layers becoming too thin, and (3) a local smoothing objective from Huang et al. [2023]. Fig. 6 demonstrates the simulated and fabricated results, where we notice that the optimization effectively limits movement on the face and legs of the frog while allowing the neck to inflate. This matches very closely with the behavior of the fabricated example.

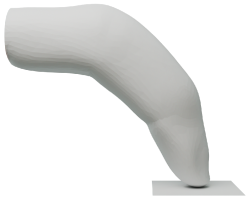


(a) Baseline chamber

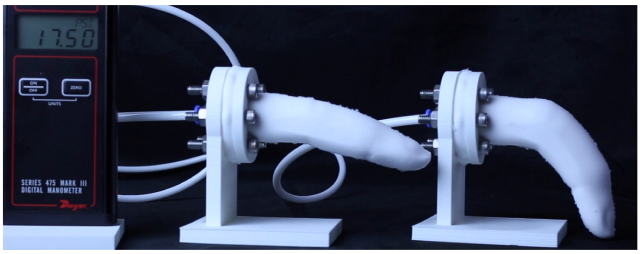
(b) Optimized chamber



(c) Simulated baseline inflation



(d) Simulated optimized inflation

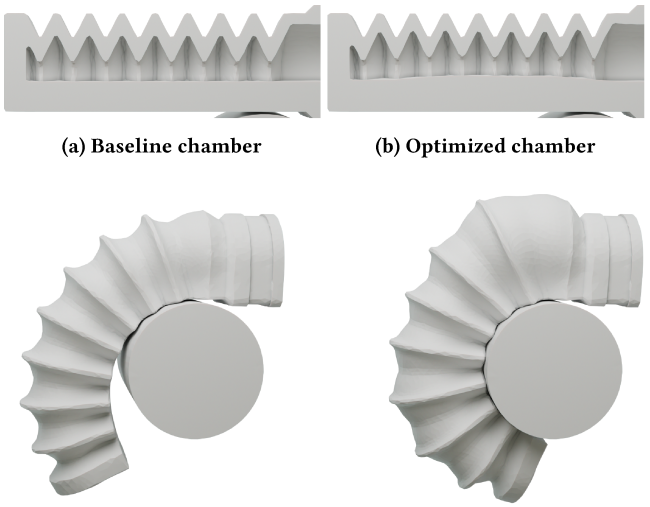


(e) Experimental inflation

Figure 7: Shape optimization of the interior cavity of a finger to match an artistic target of a bent finger.

Finger. In a similar setting to the frog, we again aim to optimize an artistic target that resembles the actuation of a finger. We take a model finger and add a cavity inside, aligning the cavity with an imagined skeleton but otherwise with no prior knowledge of how it would act under inflation. We then take the surface of the finger and modify the geometry using Blender to achieve the artistic target of a finger bending. The target is far from the rest pose, displacing the finger tip by around 50% of the length of the finger. We simulate the inflation to 120 kPa (17.4 psi), and show the simulated and fabricated results in Fig. 7. The optimized design closely matches the expected target, which is remarkable considering that it is using a single pneumatic chamber and it is composed of a single isotropic material.

Gripper. We optimize an existing pneumatic gripper to maximize its ability to grip a prescribed cylinder in a dynamic simulation with contact and friction forces. The starting gripper draws heavily from existing work on grippers in literature [Mosadegh et al. 2014], however its design has not been optimized beyond a first draft. We simulate inflation to 60 kPa (8.7 psi) in a scene with a static cylinder, with contact and friction enabled. Then, optimization is carrier out to match an undersized cylinder target mesh and maximize the contact force between the gripper contact surface and the physical cylinder. The results of the optimization are fabricated and are shown in Fig. 8. The simulated total forces on the cylinder increased from $5.2N$ in the baseline to $9.5N$ in the optimized case.



(a) Baseline chamber (b) Optimized chamber

(c) Simulated baseline inflation (d) Simulated optimized inflation

(e) Experimental inflation

Figure 8: Optimization of the interior cavity of a gripper to match a target cylinder and maximize contact force with the physical cylinder.

We also compare the optimization of the gripper with and without the surface force objective to evaluate its effectiveness. We give each optimization the same machine resources and visualize the final results in Fig. 11. While both optimizations yield reasonable results in wrapping around the cylinder, the surface force objective produces a final shape with surfaces in tighter contact and a considerably higher total force exerted on the cylinder of around $9.5N$ compared to $6N$ for the target matching optimization.

Worm. Finally, we showcase shape design optimization in a contact- and friction-rich dynamic setting involving locomotion. We take a long, worm-like box with rounded corners and embed cavities running along its length. Then, the chambers are inflated to pressures dictated by phase-shifted sinusoids. This setup was inspired by biological mechanisms and the soft robots of [Shepherd et al. 2011], however we did not fabricate it due to the complexity of having many chambers with different pressures and precise control on the pressure dynamics per chamber. The original design is 50cm long and exhibits some ability for modest locomotion 20cm/min, so we attempted to optimize the shape of the cavities and the top outside surface to maximize its displacement. Since the forward simulation is expensive, we limited the simulation to the first 3 seconds and maximize the displacement in the forward direction. The optimized result is able to locomote at a rate of about 31cm/min, or around 50% faster. We show this example in Fig. 9

Figure 9: Showcase of the worm optimization. The baseline robot is on the top while the optimized one is at the bottom. This was taken after 6 seconds, or double the optimization end time.

6 CONCLUSION

Our approach demonstrates that shape optimization can be used to optimized soft robotic designs in the presence of contact and pressure forces. The high accuracy of our method, which is crucial for designing pneumatically inflated soft robots computationally, comes with the limitation of a high computational cost, especially for complex scenes involving large deformation and rich contact.

Future Directions. We have focused on offline, objective-driven pneumatic actuator design in this work, keeping actuation parameters fixed. An interesting future avenue is to incorporate actuation parameters into our design optimization, solving for optimal control parameters simultaneously [Bächer et al. 2021]. To this end, we added differentiability with respect to pressure control and optimized the locomotion of a simple 2D walker inspired by Jatavallabhula et al. [2021], shown in Fig. 12. However, validating control optimization in an experimental setup would require modeling and simulating a pressure controller, which we leave to future work.

To control soft robots, optimal sensorization is a problem that is dual to actuation, but closely related [Tapia et al. 2020]. Our ultimate goal is to provide a robust simulator for general soft robots, co-optimizing their design and actuation parameters to enable the rapid fabrication of versatile, autonomous soft robotics [Katzschmann et al. 2018; Laschi et al. 2012; Wehner et al. 2016].

ACKNOWLEDGMENTS

This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. This work was also partially supported by the NSF grants OAC-1835712, CHS-1908767, CHS-1901091, IIS-2313156, a Sloan Fellowship, and Disney Research.

REFERENCES

Moritz Bächer, Espen Knoop, and Christian Schumacher. 2021. Design and Control of Soft Robots Using Differentiable Simulation. *Current Robotics Reports* 2 (06 2021). https://doi.org/10.1007/s43154-021-00052-7

Pierre Baque, Edoardo Remelli, Francois Fleuret, and Pascal Fua. 2018. Geodesic Convolutional Shape Optimization. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 472–481. https://proceedings.mlr.press/v80/baque18a.html

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

James Bern, Pol Banzet, Roi Poranne, and Stelian Coros. 2019. Trajectory Optimization for Cable-Driven Soft Robot Locomotion. In *Robotics: Science and Systems XV*, Vol. 1. Robotics: Science and Systems Foundation. https://doi.org/10.15607/rss.2019.xv.052

James M. Bern, Kai-Hung Chang, and Stelian Coros. 2017. Interactive Design of Animated Plushies. *ACM Trans. Graph.* 36, 4, Article 80 (jul 2017), 11 pages. https://doi.org/10.1145/3072959.3073700

James M. Bern, Yannick Schnider, Pol Banzet, Nitish Kumar, and Stelian Coros. 2020. Soft Robot Control With a Learned Differentiable Model. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 417–423. https://doi.org/10.1109/robosoft48309.2020.9116011

J. Bonet, R.D. Wood, J. Mahaney, and P. Heywood. 2000. Finite element analysis of air supported membrane structures. *Computer Methods in Applied Mechanics and Engineering* 190, 5 (2000), 579–595. https://doi.org/10.1016/S0045-7825(99)00428-4

Michael Chang, Tomer Ullman, Antonio Torralba, and Joshua Tenenbaum. 2017. A Compositional Object-Based Approach to Learning Physical Dynamics. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkab5dqxe

Feifei Chen and Michael Yu Wang. 2020. Design optimization of soft robots: A review of the state of the art. *IEEE Robotics & Automation Magazine* 27, 4 (2020), 27–43.

Alban de Vaucorbeil, Vinh Phu Nguyen, Sina Sinaie, and Jian Ying Wu. 2019. Material point method after 25 years: theory, implementation and applications. *Submitted to Advances in Applied Mechanics* (2019), 1.

Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. DiffPD: Differentiable Projective Dynamics. *ACM Trans. Graph.* 41, 2, Article 13 (nov 2021), 21 pages. https://doi.org/10.1145/3490168

Zachary Ferguson, Pranav Jain, Denis Zorin, Teseo Schneider, and Daniele Panozzo. 2023. High-Order Incremental Potential Contact for Elastodynamic Simulation on Curved Meshes. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, 11 pages.

Konstantinos Gavriil, Ruslan Guseinov, Jesús Pérez, Davide Pellis, Paul Henderson, Florian Rist, Helmut Pottmann, and Bernd Bickel. 2020. Computational Design of Cold Bent Glass FaçAdes. *ACM Trans. Graph.* 39, 6, Article 208 (nov 2020), 16 pages. https://doi.org/10.1145/3414685.3417843

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 Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Olivier Goury and Christian Duriez. 2018. Fast, Generic, and Reliable Control and Simulation of Soft Robots Using Model Order Reduction. *IEEE Transactions on Robotics* 34, 6 (2018), 1565–1576. https://doi.org/10.1109/TRO.2018.2861900

Christian Hafner, Christian Schumacher, Espen Knoop, Thomas Auzinger, Bernd Bickel, and Moritz Bächer. 2019. X-CAD: Optimizing CAD Models with Extended Finite Elements. *ACM Trans. Graph.* 38, 6, Article 157 (Nov. 2019), 15 pages. https://doi.org/10.1145/3355089.3356576

David Hahn, Pol Banzet, James M. Bern, and Stelian Coros. 2019. Real2Sim: Visco-Elastic Parameter Estimation from Dynamic Motion. *ACM Trans. Graph.* 38, 6, Article 236 (Nov. 2019), 13 pages. https://doi.org/10.1145/3355089.3356548

M. Haßler and K. Schweizerhof. 2008. On the static interaction of fluid and gas loaded multi-chamber systems in large deformation finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 197, 19 (2008), 1725–1749. https://doi.org/10.1016/j.cma.2007.08.028 Computational Methods in Fluid–Structure Interaction.

Eric Heiden, Miles Macklin, Yashraj S Narang, Dieter Fox, Animesh Garg, and Fabio Ramos. 2021. DiSECt: A Differentiable Simulation Engine for Autonomous Robotic Cutting. In *Proceedings of Robotics: Science and Systems*. Virtual. https://doi.org/10.15607/RSS.2021.XVII.067

Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S Sukhatme. 2020. NeuralSim: Augmenting Differentiable Simulators with Neural Networks. *arXiv preprint arXiv:2011.04217* (2020).

Shayan Hoshyari, Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2019. Vibration-Minimizing Motion Retargeting for Robotic Characters. *ACM Trans. Graph.* 38, 4, Article 102 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3323034

Jerry Hsu, Nghia Truong, Cem Yuksel, and Kui Wu. 2022. A General Two-Stage Initialization for Sag-Free Deformable Simulations. *ACM Trans. Graph.* 41, 4, Article 64 (jul 2022), 13 pages. https://doi.org/10.1145/3528223.3530165

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. 2019a. DiffTaichi: Differentiable Programming for Physical Simulation. In *International Conference on Learning Representations*.

Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B Tenenbaum, William T Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. 2019b. Chainqueen: A real-time differentiable physical simulator for soft robotics. In *2019 International conference on robotics and automation (ICRA)*. IEEE, 6265–6271.

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

Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, and Denis Zorin. 2023. Differentiable solver for time-dependent deformation problems with contact. arXiv:2205.13643 [cs.GR]

Filip Ilievski, Aaron D. Mazzeo, Robert F. Shepherd, Xin Chen, and George M. Whitesides. 2011. Soft Robotics for Chemists. *Angewandte Chemie* 123, 8 (2011), 1930–1935. https://doi.org/10.1002/ange.201006464

Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011a. Bounded Biharmonic Weights for Real-Time Deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 30, 4 (2011), 78:1–78:8.

Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011b. Bounded Biharmonic Weights for Real-Time Deformation. *ACM Trans. Graph.* 30, 4, Article 78 (jul 2011), 8 pages. https://doi.org/10.1145/2010324.1964973

Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jerome Parent-Levesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. 2021. gradSim: Differentiable simulation for system identification and visuomotor control. *International Conference on Learning Representations (ICLR)* (2021). https://openreview.net/forum?id=c_E8kFWfhp0

C. Kane, Jerrold E. Marsden, Michael Ortiz, and Matthew West. 2000. Variational Integrators and the Newmark Algorithm for Conservative and Dissipative Mechanical Systems. *Internat. J. Numer. Methods Engrg.* 49 (2000), 1295–1325. https://api.semanticscholar.org/CorpusID:1681067

Robert Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. 2018. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics* 3 (03 2018), eaar3449. https://doi.org/10.1126/scirobotics.aar3449

Cecilia Laschi, Matteo Cianchetti, Barbara Mazzolai, Laura Margheri, Maurizio Follador, and Paolo Dario. 2012. Soft Robot Arm Inspired by the Octopus. *Advanced Robotics* 26, 7 (Jan. 2012), 709–727. https://doi.org/10.1163/156855312x626343

F. Lekien and J. Marsden. 2005. Tricubic interpolation in three dimensions. *Internat. J. Numer. Methods Engrg.* 63, 3 (2005), 455–471. https://doi.org/10.1002/nme.1296

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 Trans. Graph. (SIGGRAPH)* 39, 4, Article 50 (2020).

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2023b. Convergent Incremental Potential Contact. arXiv:2307.15908 [math.NA]

Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2023a. Neural Metamaterial Networks for Nonlinear Material Design. *ACM Trans. Graph.* 42, 6, Article 186 (dec 2023), 13 pages. https://doi.org/10.1145/3618325

Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. 2022. DiffCloth: Differentiable Cloth Simulation with Dry Frictional Contact. *ACM Trans. Graph.* (mar 2022). https://doi.org/10.1145/3527660 Just Accepted.

Junbang Liang, Ming Lin, and Vladlen Koltun. 2019. Differentiable cloth simulation for inverse problems. *Neural Information Processing Systems* (2019).

Yang Liu and Michael Yu Wang. 2014. Topology design of a conforming gripper with distributed compliance via a level set method. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. 2191–2196. https://doi.org/10.1109/ROBIO.2014.7090662

Yiyue Luo, Kui Wu, Andrew Spielberg, Michael Foshey, Daniela Rus, Tomás Palacios, and Wojciech Matusik. 2022. Digital Fabrication of Pneumatic Actuators with Integrated Sensing by Machine Knitting. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 175, 13 pages. https://doi.org/10.1145/3491102.3517577

Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. *ACM Trans. Graph.* 37, 6, Article 201 (dec 2018), 16 pages. https://doi.org/10.1145/3272127.3275036

Li-Ke Ma, Yizhonc Zhang, Yang Liu, Kun Zhou, and Xin Tong. 2017. Computational Design and Fabrication of Soft Pneumatic Objects with Desired Deformations. 36, 6, Article 239 (nov 2017), 12 pages. https://doi.org/10.1145/3130800.3130850

Juan Montes Maestre, Ronan Hinchet, Stelian Coros, and Bernhard Thomaszewski. 2023. ToRoS: A Topology Optimization Approach for Designing Robotic Skins. *ACM Trans. Graph.* 42, 6, Article 194 (dec 2023), 11 pages. https://doi.org/10.1145/3618382

Liane Makatura, Bohan Wang, Yi-Lu Chen, Bolei Deng, Chris Wojtan, Bernd Bickel, and Wojciech Matusik. 2023. Procedural Metamaterials: A Unified Procedural Graph for Metamaterial Design. *ACM Trans. Graph.* 42, 5, Article 168 (jul 2023), 19 pages. https://doi.org/10.1145/3605389

Guirec Maloisel, Espen Knoop, Christian Schumacher, and Moritz Bacher. 2021. Automated Routing of Muscle Fibers for Soft Robots. *IEEE Trans. Robot.* 37, 3 (June 2021), 996–1008. https://doi.org/10.1109/tro.2020.3043654

Guirec Maloisel, Espen Knoop, Christian Schumacher, Bernhard Thomaszewski, Moritz Bächer, and Stelian Coros. 2023. Optimal Design of Flexible-Link Mechanisms With Desired Load-Displacement Profiles. *IEEE Robotics and Automation Letters* 8, 7 (2023), 4203–4210. https://doi.org/10.1109/LRA.2023.3281289

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

Sebastian K. Mitusch, Simon W. Funke, and Jørgen S. Dokken. 2019. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software* 4, 38 (2019), 1292. https://doi.org/10.21105/joss.01292

Bobak Mosadegh, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F. Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor J. Walsh, and George M. Whitesides. 2014. Pneumatic Networks for Soft Robotics that Actuate Rapidly. *Advanced Functional Materials* 24, 15 (2014), 2163–2170. https://doi.org/10.1002/adfm.201303288

Alexander Niewiarowski, Sigrid Adriaenssens, and Ruy Marcelo Pauletti. 2020. Adjoint optimization of pressurized membrane structures using automatic differentiation tools. *Computer Methods in Applied Mechanics and Engineering* 372 (2020), 113393. https://doi.org/10.1016/j.cma.2020.113393

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (2e ed.). Springer, New York, NY, USA.

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Trans. Graph.* 36, 4, Article 122 (July 2017), 16 pages. https://doi.org/10.1145/3072959.3073649

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages.

Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. 2020. Scalable Differentiable Physics for Learning and Control. In *International Conference on Machine Learning*. PMLR, 7847–7856.

Junior Rojas, Eftychios Sifakis, and Ladislav Kavan. 2021. Differentiable Implicit Soft-Body Physics. *arXiv preprint arXiv:2102.05791* (2021).

T. Rumpel and Karl Schweizerhof. 2003. Volume-dependent pressure loading and its influence on the stability of structures. *Internat. J. Numer. Methods Engrg.* 56 (01 2003), 211 – 238. https://doi.org/10.1002/nme.561

Daniela Rus and Michael Thomas Tolley. 2015. Design, fabrication and control of soft robots. *Nature* 521 (2015), 467–475. https://api.semanticscholar.org/CorpusID:217952627

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4, Article 136 (jul 2015), 13 pages. https://doi.org/10.1145/2766926

Christian Schumacher, Espen Knoop, and Moritz Bächer. 2020. Simulation-Ready Characterization of Soft Robotic Materials. *IEEE Robotics and Automation Letters* 5, 3 (2020), 3775–3782.

Christian Schumacher, Jonas Zehnder, and Moritz Bächer. 2018. Set-in-Stone: Worst-Case Optimization of Structures Weak in Tension. *ACM Trans. Graph.* 37, 6, Article 252 (dec 2018), 13 pages. https://doi.org/10.1145/3272127.3275085

Karl Schweizerhof and Ekkehard Ramm. 1984. Displacement Dependent Pressure Loads in Nonlinear Finite Element Analyses. *Computers & Structures* 18 (07 1984), 1099–1114. https://doi.org/10.1016/0045-7949(84)90154-8

Ashesh Sharma and Kurt Maute. 2018. Stress-based topology optimization using spatial gradient stabilized XFEM. *Structural and Multidisciplinary Optimization* 57, 1 (2018), 17–38.

Robert F. Shepherd, Filip Ilievski, Wonjae Choi, Stephen A. Morin, Adam A. Stokes, Aaron D. Mazzeo, Xin Chen, Michael Wang, and George M. Whitesides. 2011. Multigait soft robot. *Proceedings of the National Academy of Sciences* 108 (2011), 20400 – 20403.

Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. 2012. Computational Design of Rubber Balloons. *Computer Graphics Forum* 31, 2pt4 (2012), 835–844. https://doi.org/10.1111/j.1467-8659.2012.03064.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03064.x

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article 82 (jul 2013), 10 pages. https://doi.org/10.1145/2461912.2461979

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 33, 4 (2014). https://doi.org/2601097.2601166

Stanisław Stupkiewicz, Jakub Lengiewicz, and Jože 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

Javier Tapia, Espen Knoop, Mojmir Mutný, Miguel A. Otaduy, and Moritz Bächer. 2020. MakeSense: Automated Sensor Design for Proprioceptive Soft Robots. *Soft Robotics* 7, 3 (2020).

Davi Colli Tozoni, Jérémie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, and Denis Zorin. 2020. A Low-Parametric Rhombic Microstructure Family for Irregular Lattices. *ACM Trans. Graph.* 39, 4, Article 101 (jul 2020), 20 pages. https://doi.org/10.1145/3386569.3392451

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

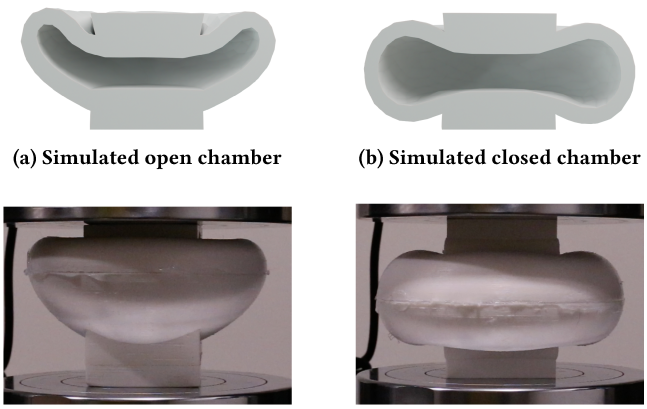
Michael Wehner, Ryan L Truby, Daniel J Fitzgerald, Bobak Mosadegh, George M Whitesides, Jennifer A Lewis, and Robert J Wood. 2016. An integrated design and fabrication strategy for entirely soft, autonomous robots. *nature* 536, 7617 (2016), 451–455.

Jie Xu, Sangwoon Kim, Tao Chen, Alberto Rodriguez Garcia, Pulkit Agrawal, Wojciech Matusik, and Shinjiro Sueda. 2022a. Efficient Tactile Simulation with Differentiability for Robotic Manipulation. In *6th Annual Conference on Robot Learning*. https://openreview.net/forum?id=6BIffCl6gsM

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. 2022b. Accelerated Policy Learning with Parallel Differentiable Simulation. https://doi.org/10.48550/ARXIV.2204.07137

Xiaoting Zhang, Xinyi Le, Zihao Wu, Emily Whiting, and Charlie C.L. Wang. 2016. Data-Driven Bending Elasticity Design by Shell Thickness. *Computer Graphics Forum (Proceedings of Symposium on Geometry Processing)* 35, 5 (2016), 157–166.

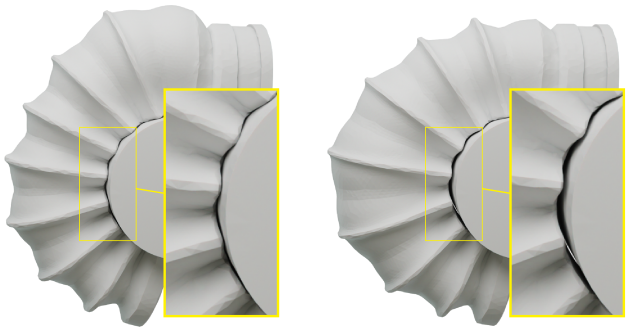
Zhan Zhang, Christopher Brandt, Jouve Jean, Wang Yue, Tian Chen, Mark Pauly, and Francis Julian Panetta. 2023. Computational Design of Flexible Planar Microstructures. *ACM Transactions on Graphics* 42, 6 (2023), 185. http://infoscience.epfl.ch/record/305985



(a) Simulated open chamber (b) Simulated closed chamber

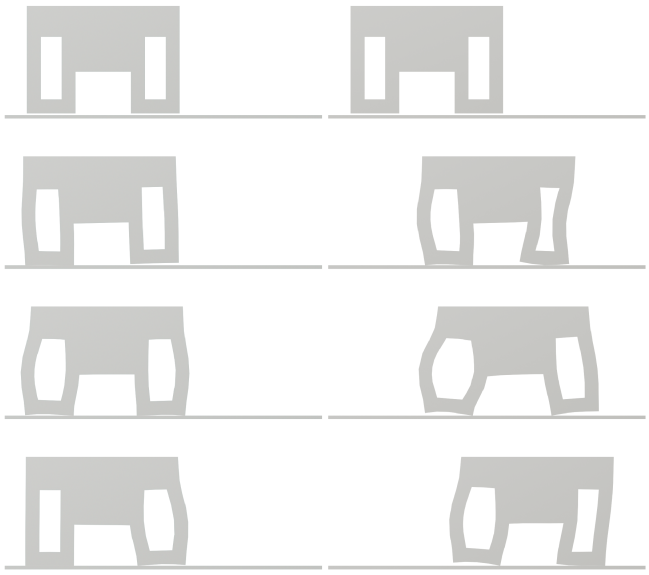
(c) Fabricated open chamber (d) Fabricated closed chamber

Figure 10: Closed chamber compression test. We model the closed chamber simulation as an adiabatic process because the silicone is a good heat insulator.



(a) Optimization of contact force (b) Optimization of shape matching

Figure 11: Evaluation of the contact force objective. In (a) we can see that the contact surfaces are closer together by optimizing force in addition to just shape in (b).



(a) Baseline walker (b) Optimized walker

Figure 12: Pressure control optimization. In (a) we can see that the starting pressure control policy produces no forward motion whereas the optimized policy (b) is very effective at moving the robot forward and keeping it upright.