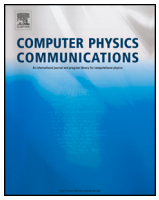
Computer Physics Communications 298 (2024) 109090

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

A rapid and automated computational approach to the design of multistable soft actuators ☆

Mehran Mirramezani *, Deniz Oktay, Ryan P. Adams

Department of Computer Science, Princeton University, Princeton, NJ, 08540, USA

ARTICLE INFO

Keywords:
Multistable structures
Soft robots
End-to-end differentiation
Optimization
Inverse design

ABSTRACT

We develop an automated computational modeling framework for rapid gradient-based design of multistable soft mechanical structures composed of non-identical bistable unit cells with appropriate geometric parameterization. This framework includes a custom isogeometric analysis-based continuum mechanics solver that is robust and end-to-end differentiable, which enables geometric and material optimization to achieve a desired multistability pattern. We apply this numerical modeling approach in two dimensions to design a variety of multistable structures, accounting for various geometric and material constraints. Our framework demonstrates consistent agreement with experimental results, and robust performance in designing for multistability, which facilities soft actuator design with high precision and reliability.

1. Introduction

Soft robots, which are comprised of compliant structures instead of rigid bodies and links, enable a wide range of smooth and complicated motions and functionalities. In particular, soft robots facilitate safe and adaptive interactions with humans and complex environments [4,43], biomedical device design [35], microsurgical instrumentation [48], manipulation of delicate objects [14], navigation through confined space [21], etc. Despite several advantages, the structural compliance of soft robots makes it challenging to achieve high-speed performance, high-precision motion control, and high-strength force output. Mechanical instability such as snap-through buckling is one approach to tackling these challenges through designing "tunable" multistable structures, enabling rapid movements and amplified force output when snapping from one stable configuration to another.

Multistability is an increasingly important aspect of compliant mechanical structures, allowing reconfigurable systems that are locally stable and so can stay in position without energy consumption [38,51]. Multistable structures are generally designed by combining several bistable structures with snap-through buckling [39], a kind of elastic instability in which a structure jumps from one stable configuration to another when external stimuli beyond critical values are applied. Initially curved beams are commonly adapted as basic elements to create such snap-through induced multistable structures (SIMS) [34,11,23], (cf. Fig. 1). In addition to ease of manufacturing via additive manufacturing, these multistable structures demonstrated promising soft robotic applications in high energy absorption and release [13,40], fast object grasping [2], and rapid/precise locomotion control in a pre-designed way by switching between stable configurations [44,22]. Fundamentally, these locally stable configurations (and the transitions between them) are determined via complex geometries [27,26] that, when combined with appropriate material choices, take advantage of highly nonlinear snap-through buckling [50].

Unfortunately, this nonlinearity and geometric complexity means that current approaches to multistablity design involve a combination of deep human expertise and experimental trial-and-error [29,41,27,26], leading to slow iteration times and suboptimal designs. Simulations via, e.g., the finite element method (FEM) [24], can in principle enable comprehensive analysis of such structures, but complex geometries and large deformations require computationally expensive simulations. This high computational cost and more importantly non-differentiable nature of most FEM solvers limited their use as a tool for optimization and inverse design of such structures, which requires computation of gradients of an objective with respect to design parameters. Although adjoint methods [10] are used as computationally effective approaches to numerically compute gradients for such problems [45,9,12], they require manually deriving derivatives for each input parameters and handling nonlinearities, discontinuities, constraints and boundary con-

☆ The review of this paper was arranged by Prof. Andrew Hazel.

* Corresponding author.
E-mail addresses: mehranmir@princeton.edu (M. Mirramezani), doktay@princeton.edu (D. Oktay), rpa@princeton.edu (R.P. Adams).

https://doi.org/10.1016/j.cpc.2024.109090
Received 15 July 2023; Received in revised form 20 December 2023; Accepted 8 January 2024
Available online 11 January 2024

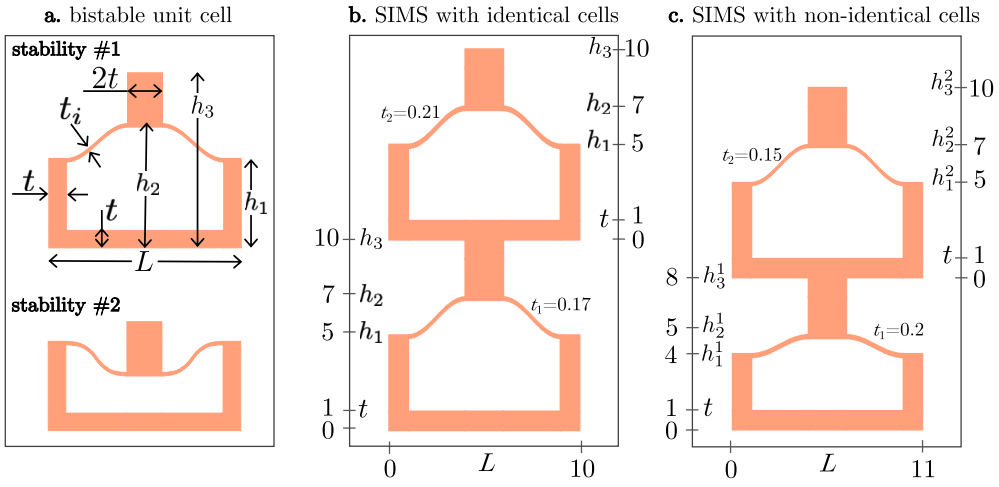


Fig. 1. a) A schematic of a snap-through bistable unit cell with geometric parameters and stable configurations. **b)** A representative SIMS constructed from two identical unit cells, but different in-plane beam thickness at each row, with seven geometric design parameters to optimize. **c)** A representative SIMS constructed from two non-identical unit cells, with 10 geometric design parameters to optimize.

ditions, making their implementation non-trivial in many problems. Therefore, FEM analysis is often employed to validate experimental observations or to understand multistable structures responses under different loading conditions or changes in geometric parameters [23,49].

Theoretical one-dimensional elastic beam models are widely utilized to mitigate these computational cost and challenges to inform designs in a more timely manner [8,11,32,47]. For example, Brenner et al., [8] used a linear beam model and an adjoint optimization technique to design a bistable switch with optimal force ratio. These models can reasonably capture qualitative behavior of multistable structures e.g., sequence of deformations and multistability [11], however, their quantitative predictions such as force amplification magnitude and stability locations could be inaccurate in many cases for a couple of reasons: First, the simplified assumptions of beam theories such as small deformation are not valid in multistability analysis with large deformations. Second, one-dimensional beam theories are based on linear-elastic constitutive models, however, soft actuators are mainly fabricated by hyperelastic materials to leverage their extensive elastic deformation and energy absorption capability. Therefore, a rapid differentiable FEM-based solver for nonlinear large-deformation modeling of hyperelastic material can significantly improve accurate optimization and inverse design strategies of soft multistable actuators.

Here we propose an automated framework that enables the rapid design of SIMS, given only a desired pattern of multistability described by a strain energy-displacement curve. The overall approach to this inverse design problem is to rapidly optimize the geometry and material property of a mechanical structure to induce a target strain energy-displacement curve, which is widely used to perform multistability analysis where its local minima determine stable configurations of the structure. First, we construct a parametric space of mechanical structures via programmatically-defined unit cells. Each unit cell of the structure has parameters determining local deformation behavior e.g., geometric parameters of a curved beam, enabling highly nonlinear macroscopic properties. We develop a differentiable solver based on isogeometric analysis (IGA) [25] that enables simulation of the statics of large-deformation continuum mechanics, while also providing the Jacobian of computed outputs with respect to input parameters and internal variables, making it possible to perform efficient optimization. We take advantage of IGA nature representing both geometry and solution in the same basis functions, which allows us to directly create a differentiable map from geometry parameters to solution basis, and also to a specified loss function (defined by the solution) using adjoint methods. Adjoint methods [10] along with automatic differentiation [3] are utilized in our framework to efficiently compute the gradients, which also alleviate customized algorithmic and implementational overhead challenges of adjoint methods. While automatic differentiation has been playing critical role in machine learning, its potential applications in various engineering problem-solving techniques have drawn more interest in recent years [31,42].

We leveraged the native differentiability and GPU acceleration of JAX [7], the successor to the Autograd tool that has been widely used in machine learning. Recently, several differentiable physics-based simulators in JAX have shown promise in other scientific computing fields, e.g., molecular dynamic simulation [37], thermodynamics modeling [18], and computational fluid dynamics simulations [30,5]. The end-to-end differentiability of our solver enables the use of modern large-scale constrained optimization techniques to directly optimize geometrical and material properties of a model to achieve desired functionality. We additionally fabricate prototype designs using additive manufacturing and compare simulation results with experiment. To the best of our knowledge, this is the only fully automated framework utilizing numerical solution of continuum mechanics for designing snap-through multistable structures with desired multistability behavior, which can make the design loop tighter and make it easier to create such structures with high-speed high-precision soft actuation capability.

2. Methods

2.1. Structural geometry

In this study, the design of multistable structure consists of multiple bistable unit cells that each includes a sinusoidal curved beam with snap-through behavior and a frame. The initial shape of the centerline of a fixed-thickness curved beam w_0 can be expressed as:

$$w_0(x) = \frac{h_2 - h_1}{2}\left[1 - \cos\frac{2\pi x}{L - 4t}\right], \tag{1}$$

where L is the length of each unit cell and t is the thickness of the frame. The out-of-plane thickness of the structure is always one cm in our modeling. Other geometric parameters and their descriptions are indicated in Fig. 1a for a representative unit cell. The beam's in-plane thickness for each cell is denoted as t_i, which is different in each row and considered a tunable geometric parameter to control the sequence of deformation in SIMS with identical unit cells [11] (Fig. 1b). This results in $5 + n$ geometric design parameters (GDP) to optimize in our framework to accomplish a desired multistability pattern for a structure with n rows, i.e. GDP $= [L, t, h_1, h_2, h_3, t_1, \cdots, t_n]$. Here we also constructed structures with non-identical cells to achieve a broader range of

multistability designs, by adding more flexibility to stability locations, that increases the number of design parameters to $2+4n$ where L and t are constant among all the cells (Fig. 1c), i.e. GDP $= [L, t, h_1^i, h_2^i, h_3^i, t_i]$ for $i = 1, \cdots, n$.

2.2. Fabrication and experiments

To validate our proposed numerical method and study multistability behavior of structures, a series of samples were fabricated via a calibrated Prusa i3 MK3S 3D printer using stl files generated automatically in our computational framework for each model. The printing material is thermoplastic polyurethanes (TPU), known as a material with excellent abrasion resistance and large elastic strain recovery performances [28]. Moreover, TPUs have a wide range of Young's modulus (E) from 10 to 10000 MPa, which is considered as a parameter in our framework to optimize the design. The value of Young's modulus for each simulation is reported accordingly in different sections. Here, we assumed a fixed Poisson's ratio value ($\nu = 0.46$) based on reported values from prior studies on 3D printed TPU materials [46]. Material properties of the experiment presented in Section 3.1 were characterized by standard tensile measurements according to ASTM D638-14 [19]. The measured Young's modulus is approximately 75 MPa.

Quasi-static uniaxial loading tests were conducted using a customized compression fixture (shown in Fig. 4e) with a 50 N tension/compression load cell and a loading rate of 23 mm/min in a displacement-control manner. Extra parts for connecting samples to the testing system and capturing appropriate fixed bottom and guided side boundary conditions were designed and printed using stiffer carbon-fiber-infused nylon.

2.3. Numerical methods

2.3.1. Overview

We develop a custom isogeometric analysis solver [25] for large-deformation plane stress statics problems, with the goal of having a robust and end-to-end differentiable simulator for multistable structure design. Given geometric design parameters, our solver will simulate the multistable structure described by the parameters, and compute the gradient (adjoint) with respect to each design parameter according to a specified downstream loss function; in our case the loss function is $\|\mathrm{SE}(D) - \mathrm{SE}_t(D)\|_2$, where $\mathrm{SE}_t(D)$ and $\mathrm{SE}(D)$ are the prescribed (target) and actual strain energy vectors of the structure computed at N displacement loading increments $D = [0, d_1, \cdots, d_{N-1}]$ that are applied at the top edge of a structure. The loss function is then minimized via gradient-based optimization technique to achieve the target strain energy-displacement response.

2.3.2. Isogeometric analysis (IGA)

IGA [25] has emerged as a powerful computational method that integrates computer-aided design (CAD) and finite element analysis (FEA) [24], offering an exact geometric representation and seamless transition from geometric modeling to numerical simulations. The fundamental principle of IGA lies in utilizing smooth basis functions employed in CAD modeling, such as non-uniform rational B-splines (NURBS), to both represent the geometry and approximate the unknown fields within the domain of interest. This avoids the need for explicit mesh generation and the associated error introduced during mesh approximation.

In IGA the geometric domain is represented as composed of P patches, each parameterized by NURBS. NURBS are built from B-splines that are defined recursively starting with piecewise constant ($p = 0$) basis functions

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \le \xi < \xi_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $\xi_i \in \mathbb{R}$ is the ith knot in a knot vector $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$ that is composed of a sequence of non-decreasing real numbers in the parametric space ξ. n is the number of basis functions and p is the polynomial order. For $p = 1, 2, 3, \cdots$, the basis functions are defined by

$$B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi). \tag{3}$$

NURBS basis functions $N_{i,p}$ can be characterized based on B-spline basis functions by assigning a positive weight w_i to each basis function

$$N_i^p(\xi) = \frac{w_i B_{i,p}(\xi)}{\sum_{k=1}^{n} w_k B_{k,p}(\xi)}. \tag{4}$$

Two dimensional (2D) NURBS basis functions with order p and q corresponding to knot vectors $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$ and $H = \{\eta_1, \eta_2, \cdots, \eta_{m+q+1}\}$, respectively, are constructed as tensor products

$$N_{i,j}^{p,q}(\xi, \eta) = N_i^p(\xi) N_j^q(\eta) = \frac{w_{i,j} B_{i,p}(\xi) C_{j,q}(\eta)}{\sum_{k=1}^{n} \sum_{l=1}^{m} w_{k,l} B_{k,p}(\xi) C_{l,q}(\eta)}. \tag{5}$$

$(\xi, \eta) \in [0, 1]$ corresponds to the knot span in IGA, which is often referred to as *parent domain*. The mapping from the parent domain to the physical domain (x, y) is created by a linear combination of NURBS basis functions, where the coefficients of basis functions are referred to as control points $\mathbf{P}_{ij}$ (analogous to nodal coordinates in FEA). Therefore, the geometry of a given patch $\boldsymbol{G}^e(\xi)$ in two dimensions (NURBS surfaces) is represented as

$$\boldsymbol{G}^e(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{P}_{ij} N_{i,j}^{p,q}(\xi, \eta). \tag{6}$$

In IGA, identical basis functions are used to represent the geometry and approximate the unknown field variable of a system of partial differential equations. As a result, the field variable within a patch i.e., displacement $\mathbf{u}^e$ can be described by

$$\mathbf{u}^e(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{U}_{ij} N_{i,j}^{p,q}(\xi, \eta), \tag{7}$$

where $\mathbf{U}_{ij}$ are control (nodal) variables, which can be computed using a Galerkin method or by finding the stationary points of a strain energy function as described in Section 2.3.4.

We visualize the decomposition of a representative bistable unit cell into 18 patches in Fig. 2a. All patches use the same B-spline basis functions with $n = m$ (number of basis functions or control points) and $p = q$ that are specified as parameters in our framework. In all of our simulations, we use piecewise quadratic B-spline basis functions ($p = 3$), and each control point represents two degrees of freedom in 2D space, i.e., x_{ij} and y_{ij}. Control points of neighboring patches have incidence constraints. Visualization of the constraints between control points in different patches, for a representative beam composed of four patches with 16 control points in each, is shown in Fig. 2b.

2.3.3. Material model

We use a nearly incompressible neo-Hookean material model [33], which is a commonly used constitutive model that can effectively capture the mechanical behavior of elastomeric materials such as TPUs typically employed in manufacturing of flexible multistable structures. The elastic properties of a neo-Hookean material are described by a hyperelastic strain energy density function. This function, $W(\mathbf{F})$, is independent of the path of deformation and is a function of the deformation gradient tensor, $F_{ij} = \partial x_i / \partial X_j$ with $\mathbf{X}$ represents the undeformed reference configuration, and $\mathbf{x}$ is the deformed current configuration. In our case, $W(\mathbf{F})$ is defined as

$$W(\mathbf{F}) = \frac{\mu}{2}(I_1 - 2 - 2\ln J) + \frac{\lambda}{2}(\ln J)^2, \tag{8}$$

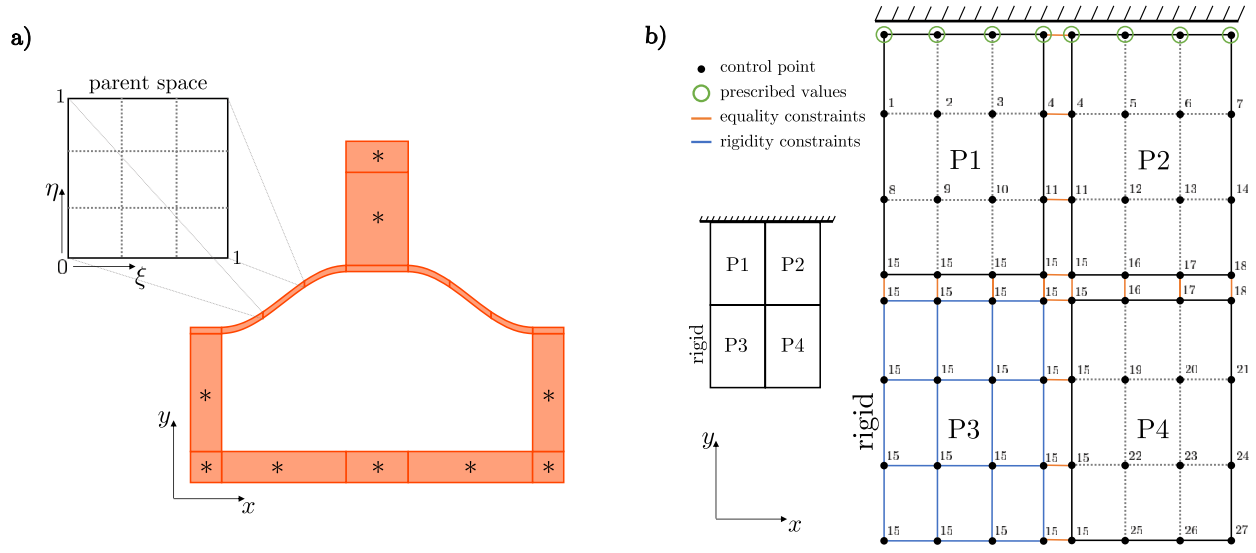


Fig. 2. a) Visualization of the decomposition of our unit cell into patches (rigid patches are starred). Each patch is pulled back into a parent space, which is a B-spline knot span. Quadrature and integration happens in this space. **b)** Visualization of the constraints between control points in different patches, for a representative beam composed of four patches with a fixed top edge. Each control point is labeled with the index it takes in the global degree of freedom array, where each control point corresponds to two degrees of freedom in 2D space. Note that even though there are 64 control points in the local array (with 128 degrees of freedom), there are only 27 control points in the global array (with 54 degrees of freedom) due to incidence constraints, Dirichlet boundary conditions, and rigidity constraints.

where $J = \det(\mathbf{F})$, $I_1 = \mathrm{tr}(\mathbf{F}^T\mathbf{F})$, and $\mu = E/2(1+\nu)$ and $\lambda = \nu E/(1+\nu)(1-2\nu)$ are Lamé parameters of a material with Young's modulus E and Poisson's ratio ν. The first term is the distortional component (independent of volume change) and the second term is the near-incompressibility constraint. We can solve for the displacement $\mathbf{u}$ by finding the stationary point of the stored energy $\Psi(\mathbf{u})$ from material mechanical deformation under external traction $\mathbf{T}$,

$$\min\left[\Psi(\mathbf{u}) = \int_{\Omega} W(\mathbf{F})d\mathbf{X} - \int_{\partial\Omega^N} \mathbf{T}\cdot\mathbf{u}d\mathbf{S}\right], \tag{9}$$

constrained by Dirichlet boundary conditions i.e., $\mathbf{u} = \mathbf{u}_b$ on $\partial\Omega^D$. We find that the energy minimization formulation dovetails extremely well with the use of automatic differentiation.

2.3.4. *Differentiable elasticity solver*

We develop an in-house end-to-end differentiable IGA framework to solve the nonlinear elasticity boundary value problem. Our implementation is entirely in JAX [7], a scientific computing and machine learning framework in Python with automatic differentiation and compilation to GPUs.

2.3.4.1. Brief introduction to automatic differentiation Automatic Differentiation (AD) is a way to systematically compute mathematical derivatives of a function in a computer program. The basic idea is to write the function by composing primitive operations in an AD framework, such as JAX [7]. These primitives each have known pre-implemented Jacobians, and are fundamental enough (e.g. matrix-vector multiplication, e^x) to be composed into functions that express most operations used in scientific computing. The derivatives are formed by tracing the primitive operations and combining their Jacobians according to the chain rule of calculus. When used correctly, AD computes exact derivatives and has low overhead; if a function takes time T to compute, the gradient and function together can be computed in time $4T$, the *Cheap Gradient Principle* [17]. This constant could be less if the gradient function is precompiled, as is supported in JAX.

The fundamental operations in JAX used for differentiation are JAX.VJP and JAX.JVP. The former computes vector-Jacobian products with a multi-input multi-output function, and the latter computes Jacobian-vector products. Concretely, given a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian is a matrix-valued function $\mathcal{J}[f] : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. A VJP is a function of input $x \in \mathbb{R}^n$ and an *adjoint* vector $v \in \mathbb{R}^m$ which computes $[\mathcal{J}[f](x)]^T v \in \mathbb{R}^n$. Note that in the case where f is a scalar-output function, the VJP exactly corresponds to the gradient (when the adjoint $v = 1$). A JVP on the other hand is a function of input $x \in \mathbb{R}^n$ and a *tangent* vector $v \in \mathbb{R}^n$ and computes $[\mathcal{J}[f](x)]v \in \mathbb{R}^m$. Note that in both cases, the full Jacobian is never necessarily formed; most AD frameworks will compute these as implicit matrix-vector products. They produce exact results to numerical accuracy, and are cheap to compute (comparable to runtime of original function).

We could also combine these fundamental operations to form a Hessian-vector product with the Hessian of a scalar-output function, again in time proportional to T, the time for function evaluation [17]. We use this fact extensively in our JAX-based solver when we use second-order methods to minimize the stored energy of the material for mechanical deformation computation.

2.3.4.2. Numerical forward solve In our framework, we solve the boundary value problem by directly optimizing the energy as a function of displacement $\mathbf{u}$, subject to prescribed boundary conditions. We define a function in JAX to compute $\Psi(\mathbf{u})$, and optimize this using a second order optimization technique, such as Newton's method. The Hessian and gradient are computed using the built-in automatic differentiation engine of JAX, with special regards to the sparsity pattern of the Hessian.

Concretely, given $\mathbf{u}$ representing the displacement of the body, we compute the strain energy $\Psi(\mathbf{u})$ described in Equation (9) with primitive JAX operations. We then call the standard library function JAX.GRAD (built off of JAX.VJP) to efficiently get the gradient function with respect to $\mathbf{u}$. We could then apply standard gradient descent (explicit) methods to optimize Ψ with only first-order information, but we find that second order (implicit) methods are needed for practical fast optimization.

Using the standard library function JAX.HESSIAN will compute the full dense Hessian, which would be impractical to use directly since it would require n_{dof} JVPs and take $n_{\text{dof}} \times n_{\text{dof}}$ memory. Instead, we take advantage of the fact that the sparsity pattern of the Hessian is static and depends solely on the connectivity of the degrees of freedom, and use Hessian-vector products (HVPs) to probe the function to materialize the sparse Hessian. In particular, we apply the Jacobian coloring algorithm [36], which gives us C special vectors $v_i \in \mathbb{R}^{\text{dof}}$ such that the

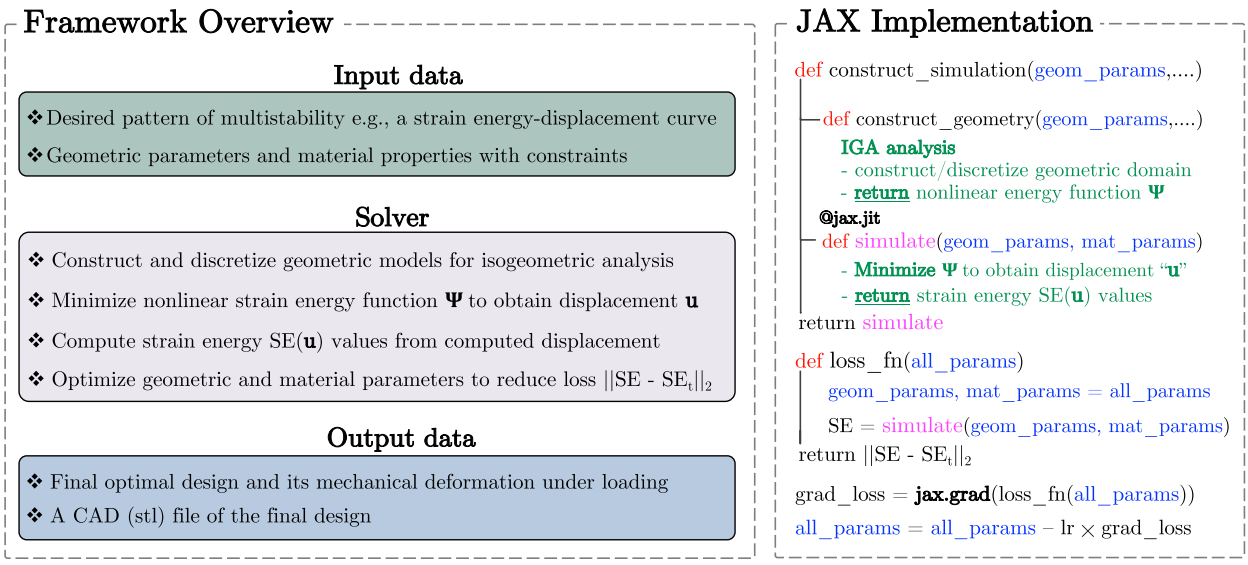


Fig. 3. An overview of the modeling steps of our automated framework implemented in JAX for multistability design.

HVPs with those vectors give us all the entries of the sparse Hessian. We then reshape those entries into a CSR sparse matrix, and use GMRES with an iLU preconditioner to solve the linear systems. For a better conditioned optimization, we use numerical continuation via load incrementing [6].

2.3.4.3. Numerical backward solve (adjoint optimization) We leverage IGA to represent both our geometry and solution in the same B-spline basis. This allows us to directly construct a differentiable map from geometry parameters to solution basis, and by using adjoint methods a differentiable map from geometry parameters to a specified loss function, enabling utilization of gradient-based optimization methods to find geometry parameters minimizing an arbitrary loss function. This differs from classic shape optimization approaches such as that in [15] as our approach will not require black-box meshing; the mapping from geometry to solution basis is well-defined and fully differentiable.

To differentiate with respect to the geometry parameters and material properties, we use the adjoint method. We wish to optimize a loss function $\mathcal{L}(\mathbf{u},\theta)$ by varying θ (representing geometry and material parameters). In addition to possible explicit dependence of $\mathcal{L}$ on θ, $\mathbf{u}$ is an implicit function of θ through the condition $\nabla\Psi(\mathbf{u};\theta)=0$, since our solution of the forward problem is a stationary point of Ψ. For simplicity, say $\theta\in\mathbb{R}^m$ and $\mathbf{u}\in\mathbb{R}^n$.

We wish to compute the total derivative $d\mathcal{L}/d\theta$. Using the chain rule

$$\frac{d\mathcal{L}}{d\theta}=\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\frac{d\mathbf{u}}{d\theta}+\frac{\partial\mathcal{L}}{\partial\theta}, \tag{10}$$

where the terms $\partial\mathcal{L}/\partial\mathbf{u}\in\mathbb{R}^n$ and $\partial\mathcal{L}/\partial\theta\in\mathbb{R}^m$ can be computed efficiently using the automatic differentiation capabilities in JAX. Meanwhile, $d\mathbf{u}/d\theta\in\mathbb{R}^{n\times m}$ is a potentially large matrix that we do not need to materialize; we are only interested in the product $\left[d\mathbf{u}/d\theta\right]^T\partial\mathcal{L}/\partial\mathbf{u}$. To compute this, note that we can use implicit differentiation on the stationarity condition, $\nabla\Psi(\mathbf{u};\theta)=0$:

$$\frac{\partial\nabla\Psi}{\partial\mathbf{u}}\frac{d\mathbf{u}}{d\theta}+\frac{\partial\Psi}{\partial\theta}=0\implies\frac{d\mathbf{u}}{d\theta}=-\left[\frac{\partial\nabla\Psi}{\partial\mathbf{u}}\right]^{-1}\frac{\partial\Psi}{\partial\theta}. \tag{11}$$

Left multiplying Equation (11) by $\partial\mathcal{L}/\partial\mathbf{u}$ gives

$$\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\frac{d\mathbf{u}}{d\theta}=-\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\left[\frac{\partial\nabla\Psi}{\partial\mathbf{u}}\right]^{-1}\frac{\partial\Psi}{\partial\theta}. \tag{12}$$

Defining $\lambda:=\partial\mathcal{L}/\partial\mathbf{u}\left[\partial\nabla\Psi/\partial\mathbf{u}\right]^{-1}$, we now have an algorithm to compute the product $\left[d\mathbf{u}/d\theta\right]^T\partial\mathcal{L}/\partial\mathbf{u}$:

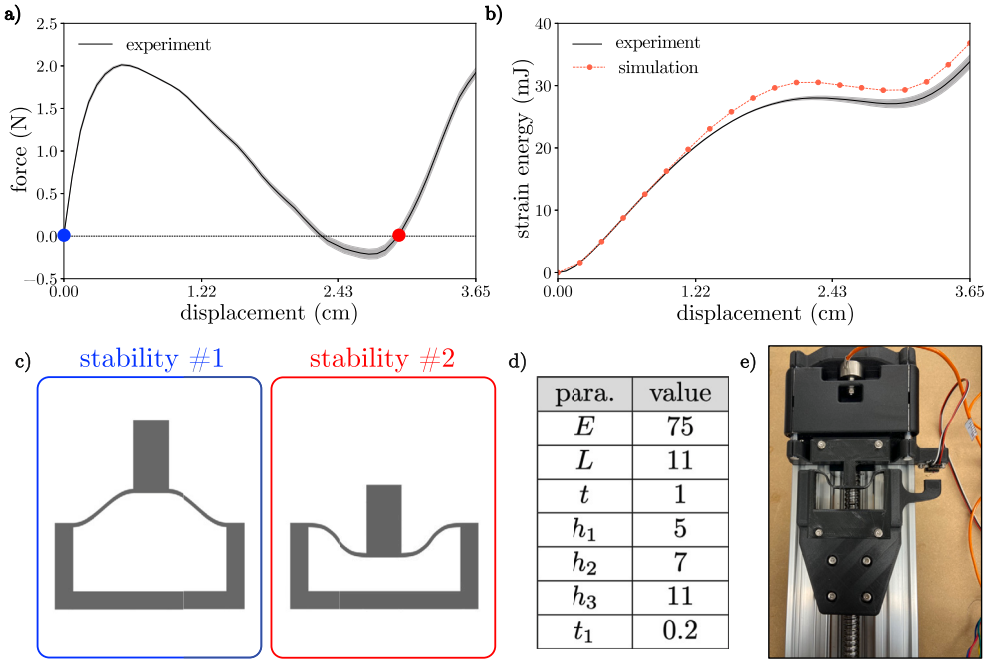
1. Compute λ as the solution to the linear system
$$\left[\frac{\partial\nabla\Psi}{\partial\mathbf{u}}\right]^T\lambda^T=\left[\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right]^T. \tag{13}$$
Note that the matrix on the LHS is exactly the Hessian of the energy at the solution point, and since the Hessian is symmetric this linear system can be solved in the same way as we did for the forward problem.
2. Compute the answer
$$\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\frac{d\mathbf{u}}{d\theta}=-\lambda\frac{\partial\Psi}{\partial\theta}. \tag{14}$$

Now that we have the total derivative (gradient) of the loss function $\mathcal{L}$ with respect to the geometric and material parameters, we can use standard first order optimization to minimize the loss function. We impose constraints on geometric and material parameters by using projected gradient descent; after each gradient update, we project into the set defined by our constraints. In our application, we only have simple box constraints, which amounts to clipping the parameters after each update.

The overview and JAX implementation of our computational framework is shown in Fig. 3. This entailed giving a desired pattern of multistability e.g., a curve of mechanical strain energy-displacement, and geometrical and material properties with constraints as the only required inputs to the framework. Given geometric and material design parameters, our solver simulates the mechanics of the structure by minimizing the nonlinear strain energy density function (forward solve), computes the gradient with respect to input design parameters and optimizes them via gradient-based optimization techniques (backward solve) to achieve the target multistability behavior. Outputs include a final optimal design, its mechanical responses under uniaxial compression loading, and an stl file of the model for 3D printing.

3. Results

Here we apply the proposed computational framework to design several bistable and multistable structures and compare the results to those derived from experiment. To ensure consistent comparisons, the same loading and boundary conditions, and material properties are used in simulations to appropriately resemble the experiments. Multiple tests were conducted for each sample to get average experimental load-displacement curves, the integral of which give strain energy values using Simpson's rule. In all simulations, the geometric parameters



para.	value
E	75
L	11
t	1
h_1	5
h_2	7
h_3	11
t_1	0.2

Fig. 4. a) Experimentally measured forces against vertical displacement for a unit cell under uniaxial loading. **b)** Elastic strain energies versus displacement from simulation and experiment. **c)** Stable configurations of the unit cell. **d)** Geometric parameters (in cm) and material property (in MPa) of the model. **e)** Our experimental setup to perform bistability test.

and the Young's modulus are optimized to reach a loss value less than 0.01. Mesh independence analysis was performed to ensure simulation results were not affected by insufficient mesh resolution, where generally seven control points for each patch was observed to be enough. In this study, all numerical results are obtained by applying a uniform compression displacement loading at the top edge of the structure.

3.1. Bistability analysis of a unit cell

Fig. 4a shows experimentally measured forces against vertical displacement for a unit cell with geometric and material properties given in Fig. 4d. The bistability deformation behavior is predicted by the experiment. The force initially increases along with the vertical displacement until a maximum critical compression force is reached, from which point snap-through instability of the structure is triggered and negative stiffness is observed. The force reduces to zero, and then becomes negative representing tension.

The elastic strain energy versus displacement curves are generally utilized for stability analysis of multistable structures, where the global and local minima of the strain energy curve are indicative of stable states of a structure. The variation of strain energy during the loading process is demonstrated in Fig. 4b from experiments and simulation. The strain energy values are the areas under the corresponding force-displacement curve from the experiment, and are computed in our isogeometric analysis. There is a global minimum at $x = 0$ and a local minimum at $x = 2.71$ in Fig. 4b, representing two stable positions of the single unit cell shown in Fig. 4c that are identically predicted by both the experiment and simulation. The local maximum energy is an unstable equilibrium position that defines the amount of energy barrier to overcome to switch from first stability to the second. Once the hill of the barrier is passed, the structure automatically jumps into a lower-energy stable configuration by releasing the additional energy.

The quantitative discrepancies between experiment and simulation results generally stem from approximated constitutive modeling assumptions, material properties estimation, and difficulties in experimentally reproducing exact boundary conditions used in simulations.

Table 1
Geometric parameters of a single unit cell model (in cm) used to create the target strain energy-displacement curve in Fig. 5.

Parameter	L	t	h_1	h_2	h_3	t_1
value	11.34	1.24	4.15	6.28	10.17	0.28

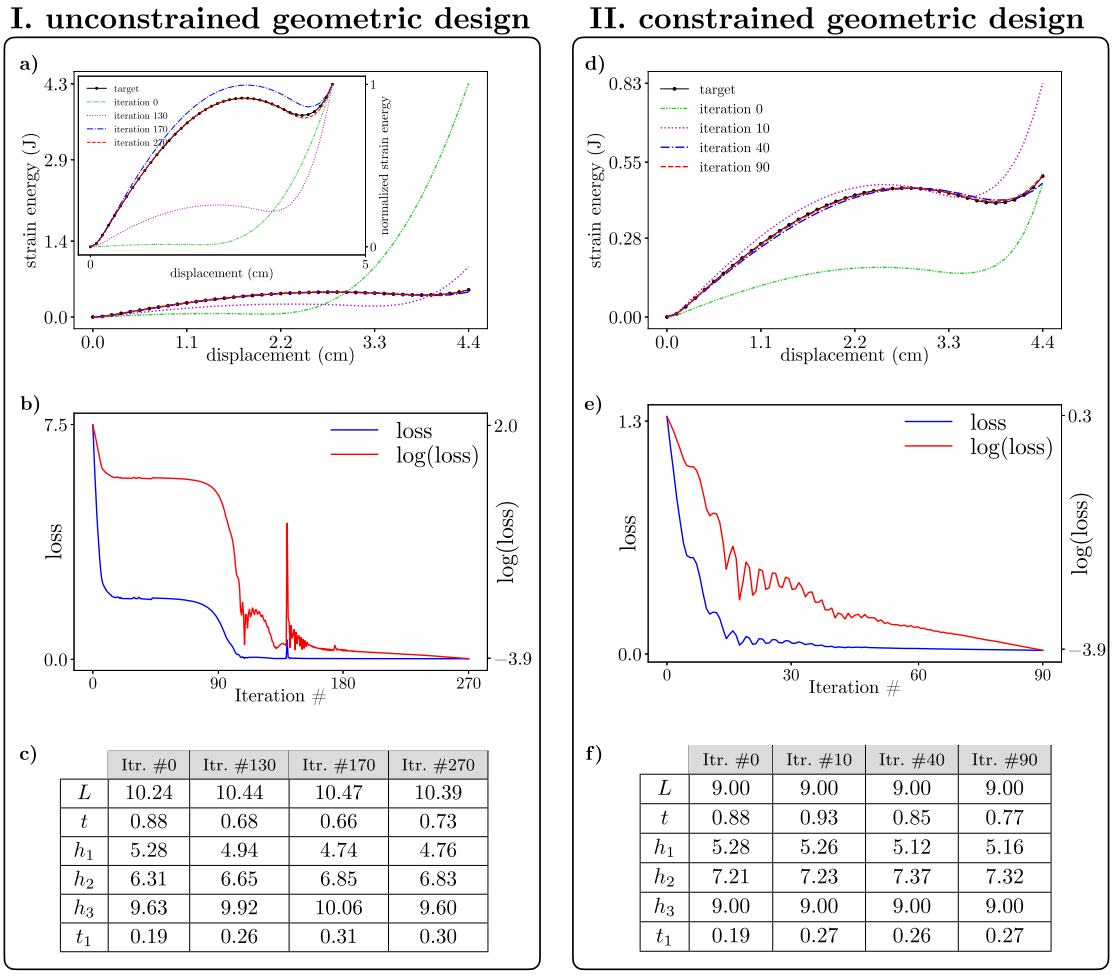
3.2. Geometric optimization for bistability design

Here, we study the ability of our framework to optimize geometric parameters (while assuming a constant material properties i.e., $E = 70$ MPa) to design a single unit cell with a desired pattern of bistability determined by a strain energy-displacement curve, which is obtained from simulating uniaxial compression loading of a unit cell with geometric parameters given in Table 1.

To evaluate the robustness of our framework, we tested two different strategies: **I)** Initial geometric parameters of the design structure (cf. Iteration #0 at Fig. 5c) are selected such that the initial strain energy-displacement curve remarkably deviates from the target curve as shown in Fig. 5a. Even with such a substantial discrepancy between initial and target strain energy values, our framework was able to find a bistable structure with a strain energy-displacement curve in agreement with the target curve after a reasonable number of iterations. **II)** Geometric design constraints are used while matching a target strain energy-displacement curve. For example, here we used same geometric parameters as in strategy **(I)** except h_2 is carefully chosen to make the stability location of the initial design close to the desired value while we are interested in a unit cell with $L = 9$ and $h_3 = 9$ (Fig. 5f). Our framework was able to design a bistable structure with these geometric constraints that has desired mechanical responses under the applied compression loading condition.

3.3. Multistability analysis and SIMS design

Multistability is studied here for a SIMS that includes three identical unit cells in series with varying in-plane beam thicknesses. Fig. 6a illus-



	Itr. #0	Itr. #130	Itr. #170	Itr. #270
L	10.24	10.44	10.47	10.39
t	0.88	0.68	0.66	0.73
h_1	5.28	4.94	4.74	4.76
h_2	6.31	6.65	6.85	6.83
h_3	9.63	9.92	10.06	9.60
t_1	0.19	0.26	0.31	0.30

	Itr. #0	Itr. #10	Itr. #40	Itr. #90
L	9.00	9.00	9.00	9.00
t	0.88	0.93	0.85	0.77
h_1	5.28	5.26	5.12	5.16
h_2	7.21	7.23	7.37	7.32
h_3	9.00	9.00	9.00	9.00
t_1	0.19	0.27	0.26	0.27

Fig. 5. a,d) Strain energy-displacement curves against displacement from uniaxial loading simulations of a designed unit cell with geometric parameters at different optimization iterations. **b,e)** Loss values versus iteration numbers. **c,f)** Geometric properties of the model (in cm) at different iterations.

Table 2
Geometric parameters of a single column SIMS model (in cm) used to perform computational multistability analysis shown in Fig. 6, and to create the target strain energy-displacement curve in Fig. 7.

Parameter	L	t	h_1	h_2	h_3	t_1	t_2	t_3
value	12.21	1.25	5.32	7.24	11.45	0.21	0.23	0.19

trates negative stiffness and tensile forces observed three times during loading process in the simulation. The presence of a global minimum, representing the initial undeformed configuration, and three local minima in the computed strain energy landscape shown in Fig. 6b indicates the multistability of this structure. The sequence of deformation the structure switches between stable configurations under uniaxial loading predicted by simulation is consistent with previous studies proposing the rows collapse in the order that thickness values increase [11]. Therefore, according to thickness values in Table 2 (the unit cells are ordered from bottom to top), the second stability corresponds to when only the top cell was collapsed followed by the third stable configuration that is achieved when top and bottom cells were collapsed (Fig. 6c). Finally, the last stability happened by applying ~ 13.5 (N) compression force required for collapsing all three cells.

The strain energy curve in Fig. 6b is now used as a desired pattern of multistability in designing a new SIMS with $L \leq 10$, $h_3 \leq 10$, and $t_i \leq 0.2$ (cf. Table 2). The eight initial geometric parameters of the model are given in Fig. 7c. Our framework optimized these geometric parameters in 92 iterations to design a structure that satisfies given geometric design constraints, and has the target strain energy-displacement curve (Fig. 7a).

We also evaluated the utility of our computational framework for designing a more complex SIMS with three non-identical unit cells resulting in 14 geometric design parameters to optimize. The desired multistablity pattern in Fig. 8a includes four stable locations at $x = 0, 4.03, 6.61, 10.35$ (cm), and energy barriers to switch from $S_1 \rightarrow S_2$, $S_2 \rightarrow S_3$, and $S_3 \rightarrow S_4$ are 0.15, 0.11, and 0.18 (J), respectively. Fig. 8b demonstrates the deformation sequence of multistability pattern for the final design with given geometric parameters in Fig. 8c, i.e., the bottom row collapses first, then the middle row and finally the top row.

Here, our framework is utilized to simultaneously optimize geometric parameters and Young's modulus of a material to build a SIMS with particular multistability behavior. This enables designing a desired multistability pattern with specific material choices. We examined this capability of our framework for a SIMS with two non-identical unit cells to achieve a target multistability design in Fig. 9a. The geometric parameters and material property used to create the target strain energy-displacement curve are presented in the first row of Table 3. We assumed that we have three material choices with $E = 20, 40$, and 80 MPa and no constraints on geometry in our design. The framework proposed a structure that has the desired multistability behavior with geometric parameters given in Fig. 9b using a material with $E = 40$ MPa, where the target strain energy-displacement curve was created assuming an $E = 60$ MPa.

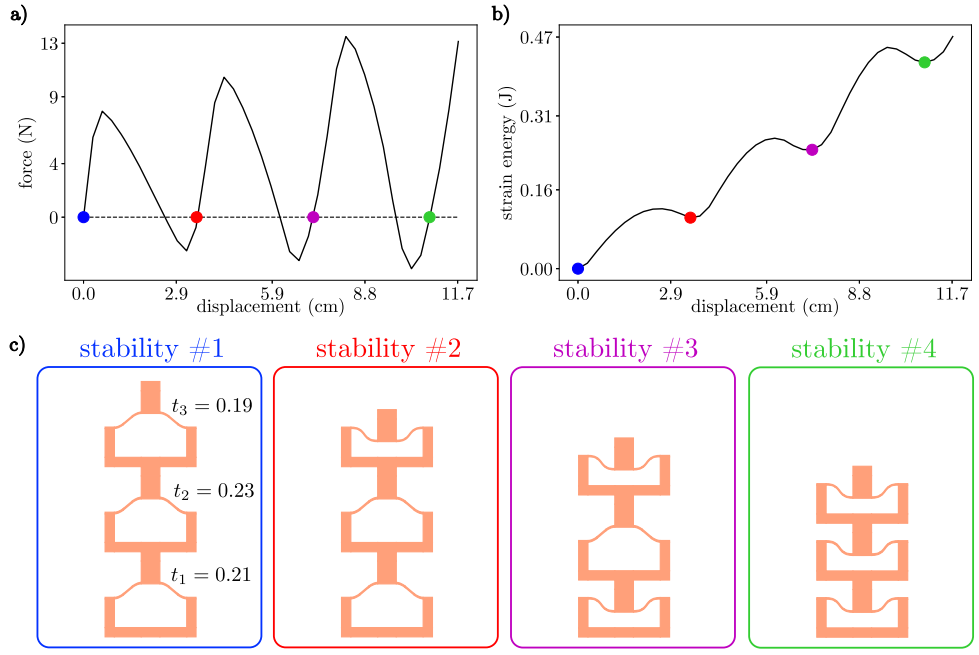
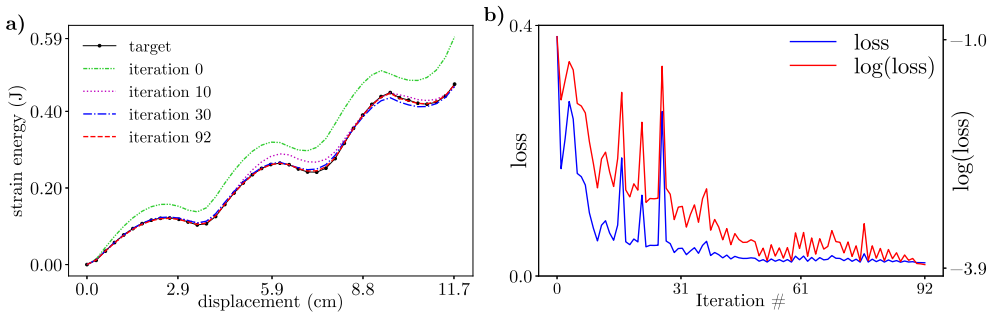


Fig. 6. a) Computed forces against vertical displacement from uniaxial loading of a single column SIMS composed of three identical unit cells with different in-plane beam thicknesses. **b)** Elastic strain energies versus displacement from simulation. **c)** Four stable configurations of the single column SIMS.



c)

	L	t	h_1	h_2	h_3	t_1	t_2	t_3
Itr. #0	10.00	1.00	4.00	5.90	10.00	0.210	0.20	0.190
Itr. #10	9.99	0.99	3.95	5.95	9.98	0.199	0.20	0.174
Itr. #30	9.97	1.01	3.97	5.94	9.99	0.189	0.20	0.173
Itr. #92	9.91	1.07	3.95	5.95	10.00	0.184	0.20	0.168

Fig. 7. a) Strain energy-displacement curves against displacement from uniaxial loading simulations of SIMS with geometric parameters at different optimization iterations. **b)** Loss values versus iteration numbers. **c)** Geometric properties of the model (in cm) at different iterations.

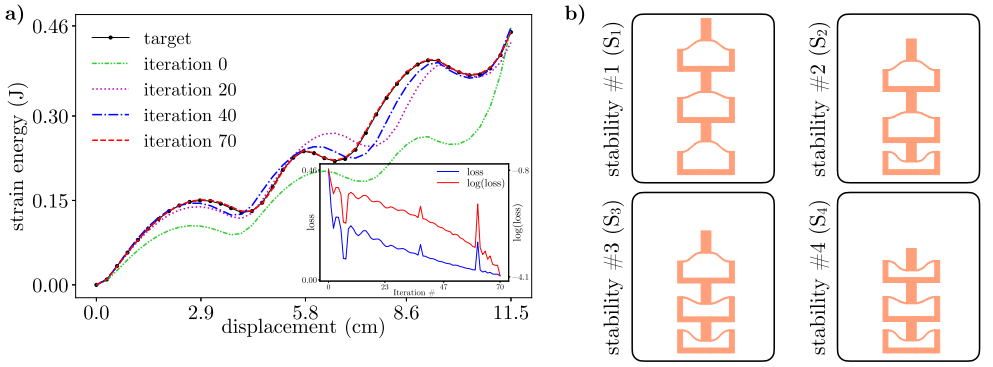
Table 3
Young's modulus (in MPa) and geometric parameters (in cm) of four different SIMS under uniaxial loading with identical strain energy-displacement curve responses shown as the target curve in Fig. 9a.

*Target curve	Geometric parameters									
Young's Modulus (E)	L	t	h_1^1	h_1^2	h_2^1	h_2^2	h_3^1	h_3^2	t_1	t_2
60*	12.00	1.00	5.00	5.00	7.00	6.30	10.00	10.00	0.230	0.210
40	11.97	1.03	5.01	5.00	6.99	6.30	10.01	10.03	0.263	0.241
20	11.88	1.11	5.02	5.01	6.99	6.29	9.89	10.44	0.324	0.298
100	12.05	0.95	4.99	5.00	7.01	6.30	9.94	9.94	0.196	0.181

In another experiment, we investigated the possibility of creating the same target strain energy curve in Fig. 9a using materials with Young's modulus of 20 MPa and 100 MPa and initial geometric parameters in the first column of Fig. 9b. The geometric parameters of the final design with identical strain energy curves for each material are presented in Table 3.

4. Discussion

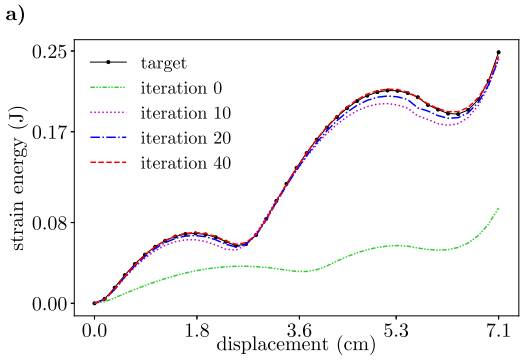
We have presented a framework implemented in JAX to rapidly design multistable mechanical structures. The proposed end-to-end differentiable IGA-based continuum mechanics solver enables geometric and material optimization of a structure to achieve a desired multista-



c)

	L	t	h_1^1	h_1^2	h_1^3	h_2^1	h_2^2	h_2^3	h_3^1	h_3^2	h_3^3	t_1	t_2	t_3
Itr. #0	12.00	1.00	5.00	5.00	5.00	7.01	6.87	6.29	11.00	11.00	11.00	0.190	0.200	0.210
Itr. #20	11.99	1.01	4.95	4.98	4.89	7.06	6.89	6.41	11.08	10.98	11.04	0.205	0.218	0.239
Itr. #40	11.98	1.02	4.99	5.08	4.79	7.03	6.78	6.50	11.03	10.94	11.09	0.211	0.212	0.237
Itr. #70	11.97	1.03	4.89	5.30	4.64	7.12	6.57	6.65	11.02	10.84	11.14	0.208	0.234	0.226

Fig. 8. a) Strain energy-displacement curves against displacement from uniaxial loading simulations of SIMS with geometric parameters at different optimization iterations, and loss values versus iteration numbers. **b)** Four stable configurations of the final designed SIMS. **c)** Geometric properties of the model (in cm) at different iterations.



b)

	Itr. #0	Itr. #10	Itr. #20	Itr. #40
E	20	40	40	40
L	11.00	11.97	11.98	11.97
t	1.00	1.02	1.03	1.03
h_1^1	4.50	5.00	5.00	5.01
h_1^2	4.50	5.03	5.01	5.00
h_2^1	6.45	7.01	7.00	6.99
h_2^2	5.74	6.27	6.29	6.30
h_3^1	11.00	10.02	10.01	10.01
h_3^2	11.00	10.03	10.02	10.03
t_1	0.190	0.257	0.260	0.263
t_2	0.20	0.238	0.239	0.241

Fig. 9. a) Strain energy-displacement curves against displacement from uniaxial loading simulations of a SIMS with geometric parameters at different optimization iterations. **b)** Geometric properties (in cm) and Young's modulus (in MPa) of the model at different iterations.

Table 4
Compilation and execution times (in seconds) from simulating single column SIMS with different number of rows under uniaxial compression loading (100 simulations for each).

Computational time (s)	Number of rows				
	one	two	three	four	five
compilation	142.90 ± 1.18	162.33 ± 2.32	191.85 ± 2.45	223.68 ± 3.53	298.56 ± 3.21
execution	7.24 ± 1.50	16.39 ± 1.18	34.64 ± 3.26	51.46 ± 3.76	109.07 ± 5.55

bility pattern. Our framework can significantly facilitate and automate soft robotic design for high precision motion control (via controlling stability locations and their sequences) and optimal force/energy output (via controlling energy barriers and releases). The inputs to the framework are geometrical and material properties, specification of boundary conditions, and a desired multistability pattern. The associated computational costs on a single GPU for compilation and execution times at each optimization iteration are shown in Table 4 for multistable structures with different number of rows, which are obtained by simulating uniaxial compression displacement loading for 100 structural models in each case. In all GPU experiments, we run on a single NVIDIA GeForce RTX 2080 Ti, while all CPU experiments we run on a Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz.

In addition to GPU acceleration that makes simulations approximately five times faster, we considered several improvements in our computational framework to speed up the simulation process. Namely, in our isogeometric analysis some of the patches within each unit cell are regarded as rigid (nine patches with star in Fig. 2a resulting in significant reduction in number of degrees of freedom (see Table 5), while strain energy values trivially change and stable locations remain the same to simulations with all deformable patches. Moreover, simulations with rigid patches generally converge with larger displacement increment size (applied on top edge), leading to reduced total computation time as shown in Table 5. Finally, using backtracking line search to determine appropriate step-size for Newton's method and applying an adaptive loading increment size technique enhanced both computational efficiency and automation of our framework.

Our framework enables reasonable flexibility in satisfying different design criteria through easily defining custom loss functions and imposing appropriate mechanical, geometrical, and material constraints. For instance, the type of multistable structures studied in this work is broadly used as soft actuators for high energy absorption and precise

Table 5
Total simulation time and execution times at each optimization iteration (in seconds), number of degrees of freedom (DoF) for modeling with and without rigid patches, and relative error values in computed strain energies (SEs) from simulating single column SIMS with different number of rows under uniaxial compression loading by enforcing geometric constrains $L = 10, h_3 = 10$, and $t_i \leq 0.2$. Error values between simulations with rigid patches and all deformable patches are computed as $e_{SE} = (1/N) \sum_{i=1}^{N} |SE_i^R - SE_i^D| / \max_n(SE_n^D)$, for N number of loading increment; the error is normalized by maximum values from simulations with all deformable patches to avoid division by small values of strain energies close to zero.

# of rows	Execution time (s)			Total time (s)		Number of DoF		Error in SEs
	w/o rig.	w/ rig.	w/ rig. (CPU)	w/o rig.	w/ rig.	w/o rig.	w/ rig.	
1	14.02	7.77	43.00	429	210	1400	800	0.91%
2	41.47	22.02	104.79	970	716	2860	1601	1.19%
3	92.12	38.99	195.02	4199	1475	4320	2402	1.37%
4	172.27	65.93	289.17	29278	5572	5780	3203	1.43%

Table 6
Energy extrema and stability locations of a designed multistable structure with target energy barriers of 0.12 and 0.14 (J), and stability at $x = 0.0, 3.83, 7.65$ (cm).

Maximum SEs (J)		Minimum SEs (J)		Energy barriers (J)		Stability loc. (cm)		
0.118	0.240	0.101	0.219	0.118	0.139	0.0	3.83	7.65

motion control that are merely characterized by the amount of energy barriers and stable locations, respectively, and not necessarily by the entire strain energy curve. Therefore, we can propose a custom loss function that allows to rapidly optimize the geometry of a mechanical structure to induce local extrema of a displacement strain energy curve with respect to a reference point, which determines the amount of energy barriers, energy releases, and axial displacement achieved by switching from one stable configuration to another. Namely, instead of a strain energy curve, the inputs to the framework could be desired extrema values of strain energy and the location of its minima. This crucially gives more freedom in designing multistable structures for such applications. For example, Table 6 indicates strain energy values and the stability location of a multistable structure that satisfies the designed criteria to absorb 0.12 and 0.14 (J) energy when consecutively moving from stable locations at $x = 0.0, 3.83, 7.65$ (cm).

Although here we focused on a particular snap-through induced multistable structure, our framework can readily be adapted for other unit cell selections with appropriate geometric parameterization such as those include stepped beams [20], two camber beams [13], twisting structures [1] among others. However, one of the limitations is that the ability of the framework to create a multistablity pattern is restricted to the space of multistability that is achievable by that particular unit cell geometry. Topology optimization techniques may potentially mitigate such challenges and enable a broader range of multistability patterns design. A limitation in terms of numerical methods is that our current solver would have trouble at large numbers of degrees of freedom, especially when modeling 3D structures, due to the iLU factorization we perform to precondition the iterative solver. Future extensions include modern multigrid methods to scale our solver up when we consider 3D problems or 2D problems with very large number of degrees of freedom (100,000-1,000,000).

Finally, our framework suffers from the lack of ability to design structures with bifurcation branches in their multistability responses observed in snap-through buckling solution of soft materials under loading [16]. A critical challenge with bifurcating branches modeling is the existence of points (states) in strain energy (or force)-displacement space where multiple equilibrium paths intersect, which cannot be captured by our proposed framework—a long standing challenge in nonlinear finite element analysis involves the identification and tracking of multiple bifurcation solution branches. Therefore, our framework can design multistable structures that only satisfy a single desired strain energy-displacement path, and not multiple solutions at bifurcation branches under the specified load.

Code availability

Our implementation is available at multistability source code.

CRediT authorship contribution statement

Mehran Mirramezani: Conceptualization, Formal analysis, Funding acquisition, Methodology, Software, Visualization, Writing – original draft. **Deniz Oktay:** Formal analysis, Methodology, Software, Writing – review & editing. **Ryan P. Adams:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgements

We would like to thank E Medina for assisting with experiments. This work was partially supported by NSF grants IIS-2007278 and OAC-2118201, the NSF under grant number 2127309 to the Computing Research Association for the CIFellows 2021 Project, and a Siemens PhD fellowship.

References

[1] A.F. Arrieta, V. v. Gemmeren, A.J. Anderson, P.M. Weaver, Dynamics and control of twisting bi-stable structures, Smart Mater. Struct. 27 (2) (2018) 025006.
[2] R. Baumgartner, A. Kogler, J.M. Stadlbauer, C.C. Foo, R. Kaltseis, M. Baumgartner, G. Mao, C. Keplinger, S.J.A. Koh, N. Arnold, Z. Suo, M. Kaltenbrunner, S. Bauer, A lesson from plants: high-speed soft robotic actuators, Adv. Sci. 7 (5) (2020) 1903391.
[3] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018) 1–43.
[4] S. Becker, T. Ranzani, S. Russo, R.J. Wood, Pop-up tissue retraction mechanism for endoscopic surgery, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 920–927.
[5] D.A. Bezgin, A.B. Buhendwa, N.A. Adams Jax-fluids, A fully-differentiable high-order computational fluid dynamics solver for compressible two-phase flows, Comput. Phys. Commun. 282 (2023) 108527.
[6] J. Bonet, R.D. Wood, Nonlinear Continuum Mechanics for Finite Element Analysis, 2 edition, Cambridge University Press, 2008.
[7] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, S. Wanderman-Milne, JAX: composable transformations of python+numpy programs, 2018.
[8] M.P. Brenner, J.H. Lang, J. Li, J. Qiu, A.H. Slocum, Optimal design of a bistable switch, Proc. Natl. Acad. Sci. 100 (17) (2003) 9663–9667.
[9] F. Bruckner, C. Abert, G. Wautischer, C. Huber, C. Vogler, M. Hinze, D. Suess, Solving large-scale inverse magnetostatic problems using the adjoint method, Sci. Rep. 7 (1) (2017) 1–7.

[10] Y. Cao, S. Li, L. Petzold, R. Serban, Adjoint sensitivity analysis for differential-algebraic equations: the adjoint dae system and its numerical solution, SIAM J. Sci. Comput. 24 (3) (2003) 1076–1089.
[11] K. Che, C. Yuan, J. Wu, H. Jerry Qi, J. Meaud, Three-dimensional-printed multistable mechanical metamaterials with a deterministic deformation sequence, J. Appl. Mech. 84 (1) (10 2016) 011004.
[12] S. Colburn, A. Majumdar, Inverse design and flexible parameterization of meta-optics using algorithmic differentiation, Commun. Phys. 4 (1) (2021) 65.
[13] D.M. Correa, T. Klatt, S. Cortes, M. Haberman, D. Kovar, C. Seepersad, Negative stiffness honeycombs for recoverable shock isolation, Rapid Prototyping J. 21 (2) (2015) 193–200.
[14] R. Deimel, O. Brock, A novel type of compliant and underactuated robotic hand for dexterous grasping, Int. J. Robot. Res. 35 (1–3) (2015) 161–185.
[15] J.S. Dokken, S.K. Mitusch, S.W. Funke, Automatic shape derivatives for transient PDEs in FEniCS and Firedrake, 2020, *arXiv*.
[16] A. Gillman, K. Fuchi, P. Buskohl, Truss-based nonlinear mechanical analysis for origami structures exhibiting bifurcation and limit point instabilities, Int. J. Solids Struct. 147 (2018) 80–93.
[17] A. Griewank, A. Walther, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, second edition, Society for Industrial and Applied Mathematics, USA, 2008.
[18] P.W. Guan, Differentiable thermodynamic modeling, Scr. Mater. 207 (2022) 114217.
[19] C.S. Ha, R.S. Lakes, M.E. Plesha, Design, fabrication, and analysis of lattice exhibiting energy absorption via snap-through behavior, Mater. Des. 141 (2018) 426–437.
[20] B. Haghpanah, L. Salari-Sharif, P. Pourrajab, J. Hopkins, L. Valdevit, Multistable shape-reconfigurable architected materials, Adv. Mater. 28 (36) (2016) 7915–7920.
[21] E.W. Hawkes, L.H. Blumenschein, J.D. Greer, A.M. Okamura, A soft robot that navigates its environment through growth, Sci. Robot. 2 (8) (2017) eaan3028.
[22] N. Hu, R. Burgueño, Buckling-induced smart applications: recent advances and trends, Smart Mater. Struct. 24 (2015) 063001.
[23] W. Hu, Z. Ren, Z. Wan, D. Qi, X. Cao, Z. Li, W. Wu, R. Tao, Y. Li, Deformation behavior and band gap switching function of 4d printed multi-stable metamaterials, Mater. Des. 200 (2021) 109481.
[24] T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.
[25] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng. 194 (39) (2005) 4135–4195.
[26] A. Ion, P. Baudisch, Interactive metamaterials, Interactions 27 (1) (2019) 88–91.
[27] A. Ion, L. Wall, R. Kovacs, P. Baudisch, Digital mechanical metamaterials, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 2017, pp. 977–988.
[28] T. Jaglinski, P. Frascone, B. Moore, D.S. Stone, R.S. Lakes, Internal friction due to negative stiffness in the indium–thallium martensitic phase transformation, Philos. Mag. 86 (27) (2006) 4285–4303.
[29] H.Y. Jeong, E. Lee, S. Ha, N. Kim, Y.C. Jun, Multistable thermal actuators via multi-material 4D printing, Adv. Mater. Technol. 4 (3) (2019) 1800495.
[30] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. 118 (21) (2021) e2101784118.
[31] A. Lindsay, R. Stogner, D. Gaston, D. Schwen, C. Matthews, W. Jiang, L.K. Aagesen, R. Carlsen, F. Kong, A. Slaughter, C. Permann, R. Martineau, Automatic differentiation in metaphysical and its applications in moose, Nucl. Technol. 207 (7) (07 2021) 905–922.
[32] J. Meaud, K. Che, Tuning elastic wave propagation in multistable architected materials, Int. J. Solids Struct. 122–123 (2017) 69–80.
[33] R. Ogden, Non-linear Elastic Deformations, Dover Civil and Mechanical Engineering, Dover Publications, 1997.
[34] Y.S. Oh, S. Kota, Synthesis of multistable equilibrium compliant mechanisms using combinations of bistable mechanisms, J. Mech. Des. 131 (2) (2009) 021002.
[35] C.J. Payne, I. Wamala, D. Bautista-Salinas, M. Saeed, D.V. Story, T. Thalhofer, M.A. Horvath, C. Abah, P.J. del Nido, C.J. Walsh, N.V. Vasilyev, Soft robotic ventricular assist device with septal bracing for therapy of heart failure, Sci. Robot. 2 (12) (2017) eaan6736.
[36] M.J.D. Powell, P.L. Toint, On the estimation of sparse Hessian matrices, SIAM J. Numer. Anal. 16 (6) (1979) 1060–1074.
[37] S. Schoenholz, E.D. Cubuk Jax MD, A Framework for Differentiable Physics, Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 11428–11441.
[38] S. Shan, S.H. Kang, J.R. Raney, P. Wang, L. Fang, F. Candido, J.A. Lewis, K. Bertoldi, Multistable architected materials for trapping elastic strain energy, Adv. Mater. 27 (29) (2015) 4296–4301.
[39] G. Simitses, D.H. Hodges, Fundamentals of Structural Stability, Butterworth-Heinemann, 2006.
[40] S. Sun, N. An, G. Wang, M. Li, J. Zhou, Snap-back induced hysteresis in an elastic mechanical metamaterial under tension, Appl. Phys. Lett. 115 (9) (2019) 091901.
[41] Z. Vangelatos, A. Micheletti, C.P. Grigoropoulos, F. Fraternali, Design and testing of bistable lattices with tensegrity architecture and nanoscale features fabricated by multiphoton lithography, Nanomaterials 10 (2020).
[42] A. Vigliotti, F. Auricchio, Automatic differentiation for solid mechanics, Arch. Comput. Methods Eng. 28 (3) (2021) 875–895.
[43] X.-Q. Wang, K.H. Chan, Y. Cheng, T. Ding, T. Li, S. Achavananthadith, S. Ahmet, J.S. Ho, G.W. Ho, Somatosensory, light-driven, thin-film robots capable of integrated perception and motility, Adv. Mater. 32 (21) (2020) 2000351.
[44] A. Wingert, M.D. Lichter, S. Dubowsky, M. Hafez, Hyper-redundant robot manipulators actuated by optimized binary-dielectric polymers, Proc. SPIE 4695 (2002).
[45] M. Xu, M. Wei, C. Li, H. Dong, Adjoint-based optimization of flapping plates hinged with a trailing-edge flap, Theor. Appl. Mech. Lett. 5 (1) (2015) 1–4.
[46] H. Yang, L. Ma, Multi-stable mechanical metamaterials by elastic buckling instability, J. Mater. Sci. 54 (4) (2019) 3509–3526.
[47] H. Yang, L. Ma, 1D to 3D multi-stable architected materials with zero Poisson's ratio and controllable thermal expansion, Mater. Des. 188 (2020) 108430.
[48] M. Zanaty, T. Fussinger, A. Rogg, A. Lovera, D. Lambelet, I. Vardi, T.J. Wolfensberger, C. Baur, S. Henein, Programmable multistable mechanisms for safe surgical puncturing, J. Med. Devices 13 (2) (2019) 021002.
[49] Y. Zhang, M. Tichem, F. van Keulen, Rotational snap-through behavior of multistable beam-type metastructures, Int. J. Mech. Sci. 193 (2021) 106172.
[50] Y. Zhang, Q. Wang, M. Tichem, F. van Keulen, Design and characterization of multi-stable mechanical metastructures with level and tilted stable configurations, Extreme Mech. Lett. 34 (2020).
[51] Z. Zhang, D. Chen, H. Wu, Y. Bao, G. Chai, Non-contact magnetic driving bioinspired Venus flytrap robot based on bistable anti-symmetric CFRP structure, Compos. Struct. 135 (2016) 17–22.