SoRoForge: End-to-End Soft Actuator Design

Lawrence Smith, *Graduate Student Member, IEEE*, and Robert MacCurdy

***Abstract*— We present SoRoForge, a design tool that addresses several limitations in the existing design and fabrication workflow for pneumatic soft actuators. We use implicit geometry functions to specify geometry and material distribution, a GUI-based software tool for interactive exploration of computational network representations of these implicit functions, and an automated tool for generating and displaying rapid simulation results of candidate designs. We prioritize seamless connectivity between all stages of the design and fabrication process, and elimination of steps that require human intervention. The workflow presented here integrates with existing capabilities for multimaterial additive manufacturing, and is also forward-compatible with emerging automated design techniques. The data gathered from human-interaction with this tool will be used by future automation tools to enable automated soft actuator design based on high-level specifications.**

***Note to Practitioners*—In this paper we present SoRoForge, a software tool that integrates major stages of soft actuator design and fabrication, which are ordinarily distinct and require deep domain expertise to investigate.This work promotes cross-discipline collaboration, allowing experts in other fields to map soft robotics solutions onto their design problems.Our key contribution is the automation of normally tedious or complex aspects of soft actuator design and fabrication: the specification of design geometry, the preparation of numerical simulation studies, and the physical realization of finished designs.Applications for soft actuators designed using our tool include soft manipulators on automated assembly lines, patient-specific biomedical implements, and food handling equipment for easily bruised produce.**

***Index Terms*— Soft actuators, automation, design.**

I. Introduction

SOFT actuator design requires multidisciplinary expertise and is largely guided by intuition earned over time-consuming design/build/test cycles, placing it out of reach for many non-experts [1]. In order to prevent the now decades-old field of soft actuator design from stagnating or turning inward, domain experts note the importance of finding cross-discipline soft robotics applications [2]. Rather than relying on the relatively few soft robotics researchers to discover meaningful applications in other fields, we advocate for lowering barriers to entry, so that soft actuator design is more accessible to curious experts in other fields. We also predict that re-imagining the soft actuator design process will allow researchers to bring automated design tools to bear on this design space, blending human-powered and machine-powered design in the way already common in the design of novel lightweight structures [3].

Manuscript received 27 June 2022; revised 27 October 2022 and 12 January 2023; accepted 24 January 2023. Date of publication 8 February 2023; date of current version 3 July 2023. This work was supported by the Internal Lab Startup Funds. This article was recommended for publication by Associate Editor S. Dauzère-Pérès and Editor X. Xie upon evaluation of the reviewers' comments. *(Corresponding author: Robert MacCurdy.)*

The authors are with the MACLaboratory, Department of Mechanical Engineering, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: maccurdy@colorado.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3241866.

Digital Object Identifier 10.1109/TASE.2023.3241866

In this paper, we build upon prior work [4] in re-imagining soft actuator design and introduce the open-source design tool **SoRoForge**. This tool represents the embodiment of a new design paradigm for soft actuators, and for the first time synthesizes all major stages of the design, evaluation, and fabrication process in a single tool, enabling:

- interactive exploration of implicit geometry functions represented by computational networks
- online 3D static geometry preview of candidate designs
- full integration with open-source FEA software
- *multimaterial* SoRo design
- one-click creation of 3D printable design files
- file I/O for SoRoForge designs, allowing users to save, load, and remix others' concepts

II. Existing Soft Actuator Design Paradigm

The existing design and fabrication workflow for soft robotics is characterized by manual processes and bottlenecks between design, simulation of performance, and fabrication phases [5]. This paradigm is described in detail by the *Soft Robotics Toolkit*, which was first introduced in [1] and has continued to grow in breadth, increasing accessibility to the design and fabrication of soft robots. Interactive computer aided design (CAD) is typically used to specify geometry for soft robotic actuators (and the molds required to cast them, if necessary). This requires expert attention and is often not robust to parametric revisions that produce significant morphological changes [6]. Nonlinear finite element analysis (FEA) using volumetric tetrahedral or hexahedral element meshes is typically employed to assess designs [1]. This often requires modifications to CAD geometry to simplify geometric details, manual assignment of boundary conditions, and manual mesh generation. Additionally, commercial software tools used for CAD and FEA are not open-source, hindering development of new features required for soft robotics, and may prohibitively expensive for researchers. Finally, multi-step casting processes are typically used to fabricate designs, which require manual mold design and significant effort to cast and de-mold soft robot designs [7].

Each of these steps is time-consuming, requires expert human attention, and requires a distinct software tool with with an associated embodiment of the design under development: a boundary representation for design, a computational mesh

TABLE I
SURVEY OF RELATED SOFT ACTUATOR DESIGN TOOLS

Result	Design Representation	Design Evaluation Method (Computational Cost)	Extensible to Automated Design	Integrated Simulation	Fabrication File Creation
Conventional Workflow e.g. Holland 2014 [1] *"Soft Robotics Toolkit"*	Parameterized (inflexible)	3D NL FEA (High)	No	No	No
Moseley 2016 [8]	Parameterized (inflexible)	3D NL FEA (High)	Yes	Yes	No
Caasenbrood 2018 [9] *"SoRoToki"*	Signed Distance Functions (flexible)	2D NL FEA (Medium)	Yes	Yes	No
Smith 2021 [4]	Implicit Geometry (flexible)	3D NL FEA (Low)	Yes	No	Yes
Sahibi 2021 [10] *"SoRoCAD"*	Parameterized (inflexible)	3D NL FEA (Medium)	No	Yes	Yes
Yao 2022 [11]	Parameterized (inflexible)	3D NL FEA (High)	No	Yes	Yes
This Work *"SoRoForge"*	Implicit Geometry (flexible)	3D NL FEA (Low)	Yes	Yes	Yes

Related toolkits published in recent years underline the necessity for improved soft robotics design tools, however previous results do not incorporate all aspects of design, simulation, and fabrication in one tool and provide forward-looking integration capabilities with automated design approaches. NL: Nonlinear; FEA: Finite Element Analysis.

for analysis, and a surface triangulation for fabrication [6]. Furthermore, the tools currently used for soft robotics design are choices of convenience, inherited from legacy workflows originally designed for traditional subtractive manufacturing, small deformation engineering analysis, and molding. In short, the existing soft actuator design paradigm does not serve the majority of potential soft robotics researchers, impedes wider, cross-disciplinary adoption of soft robotics, and is not forward-compatible with the future of symbiotic human/machine design. SoRoForge attempts to address these gaps.

Research published in the last few years reflects the need for a new generation of design tools (see Table I) for soft robotics. These results also point to the challenges inherent in synthesizing these tools into a single workflow while maintaining a high degree of generality. Ultimately, design tools for soft robotics should be as flexible as traditional CAD is in the design of conventional mechanical components; creating a new design tool for each emerging flavor of soft actuator offers diminishing returns.

Yao et al. [11] released an open-source GUI-based tool for the exploration of a parameterized design space for bending soft actuators, which allows users to adjust key dimensions and predict actuator behavior using finite element simulations. However the variety of designs user can explore is narrowly limited to bending soft actuators with specific geometric parameters, finite element simulations incorporated in the tool require a paid license to the commercial code *COMSOL*, and computation times are prohibitive, requiring ∼75 minutes to simulate the deformation of a typical actuator design undergoing pressurization [11]. In similar work, Moseley et al. [8] provide open source code for optimizing pre-selected dimensions in pleated pneumatic actuators, but stop short of a GUI interface and live feedback on candidate design performance. In *SoRoCAD*, Borchers et. al. [10], [12] present a software tool with similar capabilities: parameterized geometry for several types of soft actuators, one-click simulation execution using the FEA software *SOFA* [13], and automated fabrication file generation (3D-printable mold components for cast silicone actuators). While user studies [14] indicate the tool lowers design barriers for novice users, *SoRoCAD* is limited to creating a small set of geometrically similar, topologically identical actuators.

Caasenbrood maintains a growing repository [9] of design, simulation, visualization, and control tools for soft actuators, under the project *SoRoToki*. This workflow includes a novel method for performing nonlinear topology optimization on soft actuators while including material nonlinearity, large deformations, and pneumatic loads on moving boundaries [15], unique in the field [16]. However these optimizations are conducted in two dimensions, restricting the breadth of the design space to actuators exhibiting planar motions and symmetry. This work also appears to focus primarily on efficient simulation of already-designed robots for control and visualization, implementing reduced-order-modeling and idealization to one dimension [17], which enables real-time control of fabricated robots [18].

In fact, this focus on the fast simulation of *previously-designed* soft robots is common in literature - many simulation approaches developed for soft robots are created with *control* in mind, where morphology is fixed and expensive precomputation is perfectly acceptable if it enables accelerated reduced order model execution [19]. In contrast, SoRoForge is focused on *design*, and we seek to rapidly characterize novel morphologies; expensive precomputation is unacceptable.

In a precursor to this work, we [4] presented an open-source toolbox for the design, simulation, and fabrication of pneumatic soft actuators. However, this earlier work dispatches simulation jobs to the commercial finite element code *Abaqus* (©Dassault Systemes), which requires a paid license, and does

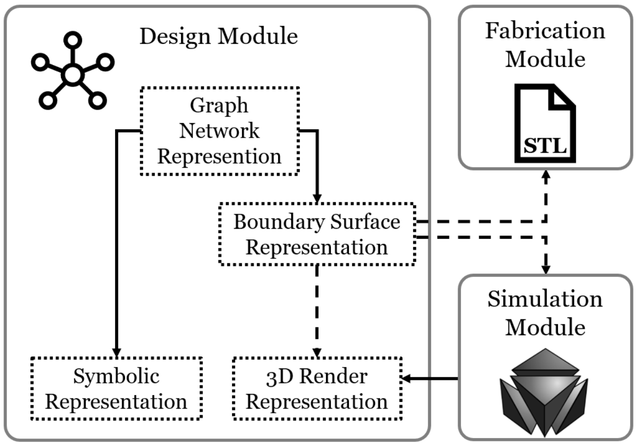

Fig. 1. Diagram depicting the seamless flow of data between SoRoForge modules that handle three phases of soft actuator creation: Design, Simulation, and Fabrication. Soft actuator designs originate as interactive graph networks (user-created or loaded from a library), terminate as ready-to-fabricate print files, and may be iteratively simulated and modified along the way. Dotted boxes indicate distinct embodiments of a single design as represented in different SoRoForge modules. Dashed arrows indicate triangular or tetrahedral remeshing operations using GIBBON [20] and solid arrows indicate conversion between representation types via graph traversal or nodal mappings. Simulation module leverages FEBio [21] for nonlinear, contact-enabled, large displacement finite element analysis.

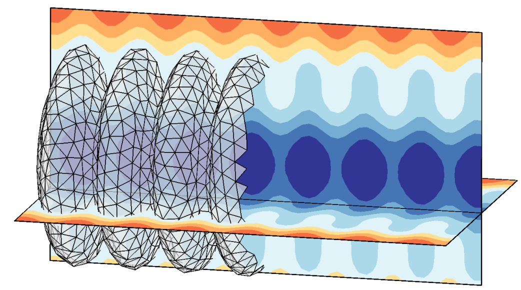

Fig. 2. Section planes through scalar field produced by an implicit geometry representation, defined by Equation 1, of a bellows-like soft actuator designed in our workflow.

not retrieve them for immediate "in-app" visualization. Also, this previous work does not provide users with interactive feedback, linking the symbolic functional form, computational network form, and deformed physical form of the designs they create, which inhibits the development of design intuition. The breadth of designs which can be realized using the previous workflow is limited by a small palette of implicit geometry "building blocks" and an implementation restriction that permits only manifold designs. Finally, the previous workflow lacks a public-facing website with tutorials and demonstrations of the tool's capabilities, and has seen limited adoption in the field.

In contrast to previous work, the workflow presented in this paper (depicted in Figure 1) prioritizes seamless connectivity between all stages of the design and fabrication process, and minimizes steps that require manual intervention. An open-source GUI-based implementation of this workflow, with all required simulation and mesh generation dependencies, is available for download at www.SoRoForge.com. We have written SoRoForge in *Matlab* (©Mathworks 2022) and leveraged the work of open-source software developers for low-level operations, including GIBBON [20] for triangular mesh refinement and plotting utilities, *TetGen* [22] for robust tetrahedral mesh generation, and FEBio [21] for nonlinear finite element simulation (see Figure 1). In the following subsections we describe our approach to design, simulation, and fabrication of soft actuators in SoRoForge, and note where our work is distinguished from other research in the field.

A. Implicit Geometry Modeling

The geometry and material distribution in many pneumatic soft actuators may be efficiently described by analytical functions in three dimensions. Because geometric features often repeat along the length of an actuator, and descriptions of geometry/material distribution in the transverse plane are natural in polar coordinates, representations of actuator geometry can be surprisingly compact. For example, the expression

$$sin(4z) - sin(2.8r) + 2.8r - 3.4 = \psi \tag{1}$$

produces the scalar field shown in Figure 2.

Triangulating an isosurface through this scalar field where $\psi = 0$ produces a surface representation of a linear actuator with rounded folds, reminiscent of the CAD-generated bellows presented in [23].

Describing geometry with implicit functions (or *signed distance functions* [24]) may appear unwieldy to a designer accustomed to building geometry using interactive CAD software. However implicit geometry modeling offers several advantages over traditional tools [24], which represent geometry by mathematically describing the surfaces and edges that bound volumes. Implicit geometry representations contain no explicit edges or surfaces - instead, functions operate on a design domain, producing a continuous scalar field. Since no references to edges or faces exist, traditionally "fragile" CAD operations such as shelling, Boolean operations, and complex filleting are reliable and near-instantaneous [25]. Additionally, the scalar field output of a geometry function contains valuable information beyond the position of the object's boundary - the sign of the field indicates whether a point is interior to the surface of the part, and the magnitude of the field gives a rough measure of the distance to the surface.

For extremely complex designs, which may contain tens of thousands of faces when represented in traditional CAD software, implicit geometry modeling excels. Implicit geometry representations typically require dramatically less storage space than traditional CAD representations and evaluation of geometry is embarrassingly parallel [26]. Commercial implicit geometry modeling platforms demonstrate 60x speed increases over traditional CAD packages when evaluating complex geometries [26]. Finally, implicit geometry functions are naturally represented by computational networks, which are compact and offer forward-looking advantages for incorporating machine intelligence into soft actuator design [16]. Previous results have noted the benefits of network-based representations of implicit geometry for automated design of soft pneumatically actuated blob-like robots [27], but our

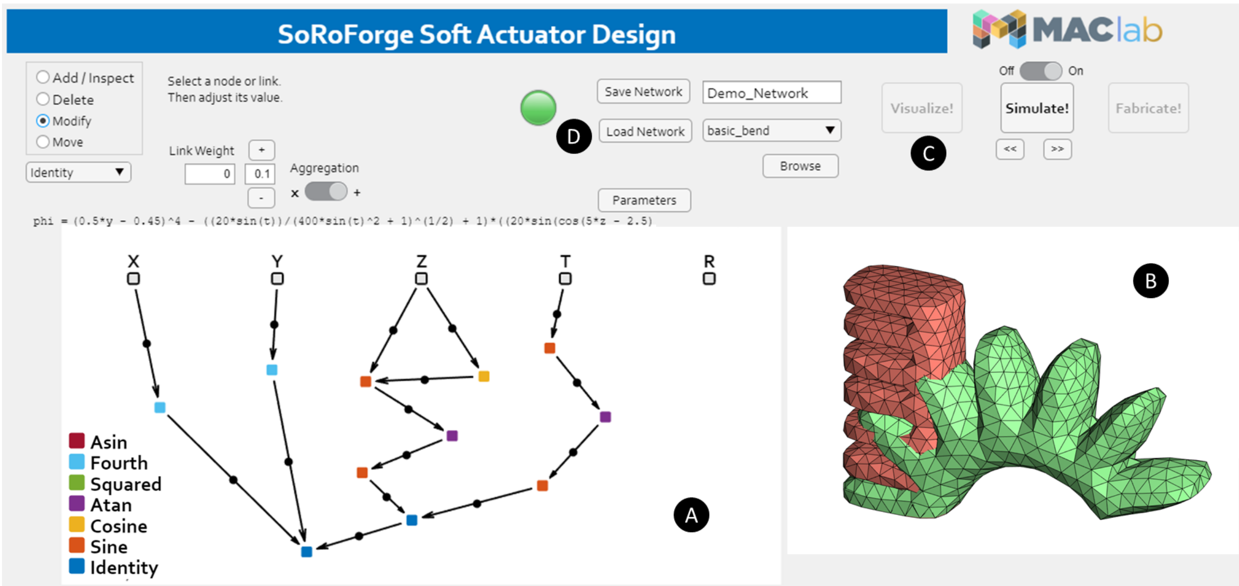

Fig. 3. SoRoForge graphical user interface for exploring multimaterial soft actuator design space. Exploration of a computational network is performed in the network editing pane A, while a rendering of the resulting actuator design is synchronously updated in the visualization pane B. Users can toggle seamlessly between design, simulation, and fabrication functions (C), while SoRoForge presents relevant information - for example the deformed state of a bending actuator in response to applied pressure, superimposed on the rest state currently visible. Users may also save out, and later load in, native SoRoForge files (D), as well as access a pre-defined library of designs including all those presented in this paper. The simulation in (B) is executed in 18s.

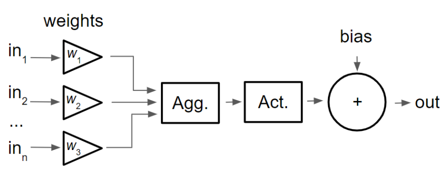

Fig. 4. Implementation of aggregation (agg.), weighting, activation (act.), and bias in the functional nodes of the computational network of SoRoForge.

TABLE II
SoRoForge Activation Functions

$sin(\cdot)$	$asin(\cdot)$	$(\cdot)^2$	$a(\cdot)$
$cos(\cdot)$	$acos(\cdot)$	$(\cdot)^4$	$a+(\cdot)$
$tan(\cdot)$	$atan(\cdot)$	$(\cdot)^{-1}$	$exp(\cdot)$

work leverages the particular elegance of descriptions of soft actuator geometry in cylindrical coordinates, and focuses directly on the design of pneumatic soft actuators.

Any implicit function may be expressed as a computational network, which readily exposes the function's structure - relationships between inputs and outputs, symmetries, repeating motifs, etc. - to an analyst. Computational networks are recognized as particularly compact and powerful generative encodings [28], and various graph and network structures have been summarized by Kicinger et al. [29]. Nodes in theses networks represent elementary functions, and links represent the flow of data between nodes. Figure 4 shows the computations inside each node of the networks we use in this work. All input signals to a node are scaled by weights associated with the links connecting them to the node, then combined into a single value using an *aggregation function*, which in our implementation can be either summation or multiplication. The aggregated value is fed into an *activation function*, which transforms the input to an output signal using one of a handful of elementary functions, such as $sin(\cdot)$, $atan(\cdot)$, and $(\cdot)^{-1}$. A full list of the activation functions we use is given in Table II.

Finally, a *bias* is applied to the signal, yielding the output value. By arranging nodes with this simple internal structure, very complex networks can be assembled, which transform coordinates into scalar field values. In our implementation, we allow computational networks to operate on the Cartesian and cylindrical coordinate representations of 3D space interchangeably, so that users can bypass tedious conversions between these systems.

B. SoRoForge Application

We present **SoRoForge** (Figure 3) for the interactive design of network representations of soft actuator geometry. Our open-source tool is built in *Matlab* and aimed to be intuitive and accessible to non-expert users. SoRoForge gives users control of basic network editing functionality such as adding and deleting nodes and links, modifying weight and bias terms, and node insertion (See Figure 3).

Computational network representations of implicit geometry functions can make interactive design more natural for a human designer than the direct manipulation of implicit geometry functions such as Equation 1. To encourage users to develop an intuitive link between an implicit function's symbolic form and its representation as a computational network, SoRoForge symbolically evaluates the graph network in real-time, simplifies the resulting expression, and displays this above the network editing pane.

As users modify network topology and weight/bias terms, SoRoForge updates a 3D visualization of the resulting undeformed actuator geometry, complete with pan/zoom/orbit controls. Users may also remesh and smooth isosurface triangulations, save and load networks, and write designs to STL files, which are naturally portable to existing 3D printers. Users may also build or load a secondary network to define the material distribution if they choose to design a multimaterial soft actuator, and write separate STL files for each distinct material. We have used this tool to recreate a variety of notable published soft actuator designs (see Figure 8). These designs were created in SoRoForge over the course of an afternoon, demonstrating the flexibility and efficiency of the tool.

C. Simulation of Candidate Designs

Numerical simulation accelerates design when it replaces or augments time-intensive design/fabricate/test/redesign iterations. However, simulation is underutilized in the design of soft robotic actuators, likely because popular simulation methods require impractical amounts of computation time and setup. This explains why simulation is overwhelmingly used for post-hoc evaluation of completed soft-robotics designs, rather than as part of the online design loop [30].

Fang et al. [31] predict the deformations of a variety of soft actuators by contriving geometric constraints to mimic actuation loads. However their work relies on the small strain assumption, limiting its usefulness in general soft actuator simulation. Their method of replacing pressure boundary conditions with volumetric "pneumatic elements" is similar to the approach of approximating pressure loads in the material point method [32]. However neither of these approaches naturally handles boundary loads, which are ubiquitous in pneumatic soft actuators, and often require calibration with empirical FEA simulation data on a per-morphology basis - making the approach slower than simply simulating via FEA when designing novel morphologies. Another approach at the cutting edge of fast physics simulation uses a neural network workflow to "learn" physics on mesh-based domains, including cloth dynamics and nonlinear structural mechanics with contact [33]. The method shows more generality and higher performance than related machine learning approaches for learning large deformation mechanics [34], and demonstrates 1-2 orders of magnitude faster solve time than the physics simulations it is trained on, but is focused primarily on accelerating simulations of roughly known geometries and boundary conditions.

Goury [35] uses model-order reduction to efficiently reduce the dimensionality of large finite element simulations of soft actuators, enabling speedups of $100x$ and real-time simulation of several complex actuators. The focus of model order reduction (MOR) is on developing highly optimized simulations of *pre-existing* fixed-morphology robots, and they require an expressive basis of deformations extracted from many full-field simulations performed offline before the MOR can be utilized (see an excellent discussion in [19]). Many results in this vein, including [35], do not report the computational cost of generating the MOR because it is worth paying in order to obtain a highly optimized model of a specific geometry for post-design tasks like control. However these costly pre-computation steps greatly limit the utility of MOR for design exploration, which is the primary focus of SoRoForge.

We note several recurring properties in soft robots that make them challenging to simulate from a numerical standpoint:

1) *Finite Strain*: common fabrication materials for soft robots create large deformations, precluding linearized small strain theory
2) *Hyperelastic, Nearly Incompressible Materials*: commonly used materials are nearly incompressible and cause stability problems and nonphysical locking effects
3) *Actuated*: pressure loads which are strongly coupled to deformation variables are common, creating additional boundary condition nonlinearities
4) *Multi-Material Constitution*: wave speeds of constituent materials span orders of magnitude, leading to stiff dynamics that challenge integration in time
5) *Intermittent, Dense Contact*: contact interactions, expensive to resolve in FEA, are expected and exploited for performance in many soft actuators
6) *Instability Phenomena*: structural instabilities are leveraged rather than avoided [36]

Key works in soft robotics literature handle the difficulty of simulating candidate designs by striking a balance between model complexity (and therefore development and execution cost) and accuracy. Table III collects significant soft robotics results with a simulation component. By far the most common approach for simulating soft robots during design is nonlinear finite element analysis (FEA) in three dimensions [1] using either tetrahedral or hexahedral volumetric finite elements, as in Table III, left column. These simulations yield results which accurately predict empirical responses for a variety of complex geometries under varied loading, albeit at substantial human and computational effort.

The preparation of a candidate actuator design for FEA requires discretizing the geometry into a *computational mesh*. Despite enormous effort, finding a general algorithm for automated generation of high-quality (numerically stable) hexahedral meshes on arbitrary geometry remains an open problem [37]; manual intervention is common. Tetrahedral meshes, which *can* be automatically generated, suffer stability issues and nonphysical locking effects when subjected to the large deformations typical of soft robots. Thus the cost of iteratively generating actuator designs and simulating them using traditional approaches is both likely to require significant human intervention and long simulation times.

Next most common in the literature are examples of dimensional reduction to 1D, yielding *beam /rod* models. Whether analytical or FE-based, these models are appropriate only for long, slender geometries, in which two spatial dimensions are much smaller than the third. These models capture the shape of a deformed actuator as well as a one-dimension model can, and are accepted as sufficiently accurate for trajectory planning and control [38]. However 1D models have significant shortcomings for design exploration; fundamental relationships between model inputs (e.g. material distribution, applied pressure) and

TABLE III
DIMENSIONAL REDUCTION IN SOFT ACTUATOR ANALYSIS

3D (unreduced)	2D (shell)	1D (beam/rod)	0D (lumped)
[39] [40] [41] [5] [42] [43] [44] [45] [46] [47]	[4] [48]	[49] [50] [51] [52] [53] [38] [54] [55] [56] [45] [57]	[28] [58]

curvature are often heuristic or require experimental data to extrapolate from, and they are suited strictly for rod-shaped geometries.

The advantage of a 1D model is exceptional reduction in computational cost in comparison to a 3D FEA model, allowing for faster than real-time simulation execution. Curiously, few examples of dimensional reduction to 2D *shell* elements can be found in literature [59] - even though this seems a natural intermediate step between expensive yet general 3D models and lightweight but limited 1D models.

SoRoForge uses shell finite elements because this method is fast, is derived from governing equations for continuum mechanics that imply physical meaning in engineering units, requires no extrapolation from empirical datasets which are costly to generate, and shares an identical mesh-based design representation with the other stages of the workflow.

Shell finite elements have only two natural coordinate dimensions and are formulated to represent geometries where one dimension of an object (the thickness) is significantly smaller than two others. Shell elements consider solution quantities constant through the thickness dimension and integrate material properties to produce zero-thickness representations of shell structures with stiffness in tension, compression, shear, and bending. Due to to their utility in structural analysis, many variations of shell finite elements have been implemented in both commercial (e.g. *Abaqus*) and open-source (e.g. FEBio [21], MOOSE [47]) software, including variants specialized for small and large deformations, multiscale and multiphysics problems, buckling, composites, and higher-order formulation [60].

Because shell finite elements have no volume, they can endure extreme deformations without loss of stability associated with element inversion and mesh entanglement, common in volumetric elements. However, this dimensional reduction comes with a cost: shell elements exhibit numerical locking, predicting artificially higher stiffness than do volumetric finite elements. Extensive study [61] has attempted to quantify this effect and offered mitigations, such as selective reduced integration (SRI) element formulations, guidelines for mesh patterns, and elements with additional bending degrees of freedom (DOF) at each node. This phenomenon is most severe in bending-dominated loading scenarios, when the shell thickness to span ratio falls below 1:10, or when element quality is poor. These assumptions are clearly violated in many soft actuator designs (such as in Figure 8). However, for the purposes of rapid evaluation of many different candidate soft actuator designs, we propose that the absolute accuracy of a simulation result should be considered *alongside* other factors such as labor cost of simulation setup, stability, and runtime.

We illustrate the utility of shell finite elements for our application by simulating the behavior of a complex tentacle-like actuator in two ways (Figure 5). At left (A) we simulate the actuator using the traditional volumetric element workflow, consistent with many published results (Table III, left column): the thickness of the actuator walls is meshed using volumetric finite elements. Because the actuator walls are thin, small tetrahedral elements must be used in order to ensure adequate element quality - though for this example we used the largest tetrahedrals allowed by the wall thickness, resulting in the lowest possible dimensionality of the resulting FE problem. In the three cases B-D, we simulate the same actuator using triangular shell elements, with the shell thickness set equal to the wall thickness of the actuator. Both simulations are executed using identical loading conditions, material properties, incrementation settings, and hardware.

The volumetric mesh contains 26*k* tetrahedral elements, with a total of 25*k* degrees of freedom, while the shell mesh contains 2*k* triangular elements and a total of 4*k* degrees of freedom. The shell-based finite element model is significantly smaller than the volumetric-based model in terms of global DOF, and contains 10*x* fewer finite elements. These factors both contribute to the difference in solve time, but the difference in element count is particularly significant. For a problem with structural nonlinearity, the global tangent stiffness matrix must be assembled many times in the full Newton-Raphson solution scheme implemented here. For the soft actuator shown in Figure 5, using a shell mesh approach decreases solve time by a factor of 5.3 relative to the volumetric mesh.

Despite the difference in solve time, the simulation results are remarkably similar: the shell-based simulation predicts the deformed shape of the actuator well (Figure 5B), and the peak displacement magnitude is within 20% of the result generated on the volumetric mesh. To further quantify the difference between the two simulation results, we compute local difference in position of the outer surface of each mesh (Figure 5C). In the undeformed configuration, we create a mapping between each node in the shell mesh and the nearest node on the surface of the volumetric mesh. At specified load increments we interpolate the deformation field of each result and then compute the difference in position of the nodes across this mapping. We normalize this distance by the length of the actuator, and report local L2 norm of the distance using color contours, showing that the maximum local difference in position predicted by the shell mesh approaches 20% of the actuator length relative to the volumetric mesh.

The artificially higher stiffness of the actuator simulated using shell elements (lower displacements at equivalent loading) is an expected result of numerical locking in shell meshes. However, we note that the trajectory predicted by the shell simulation is very similar to the volumetric mesh result, albeit requiring a larger pressure to reach a given point along that trajectory. We can improve agreement between shell and volumetric simulations by posing minimization of the L2 norm plotted in Figure 5C as an optimization problem, subject to linear scaling of the pressure applied to the shell mesh. We find that in this case a load scaling factor (LF) of 1.6 minimizes the integrated local difference in position, bringing agreement to

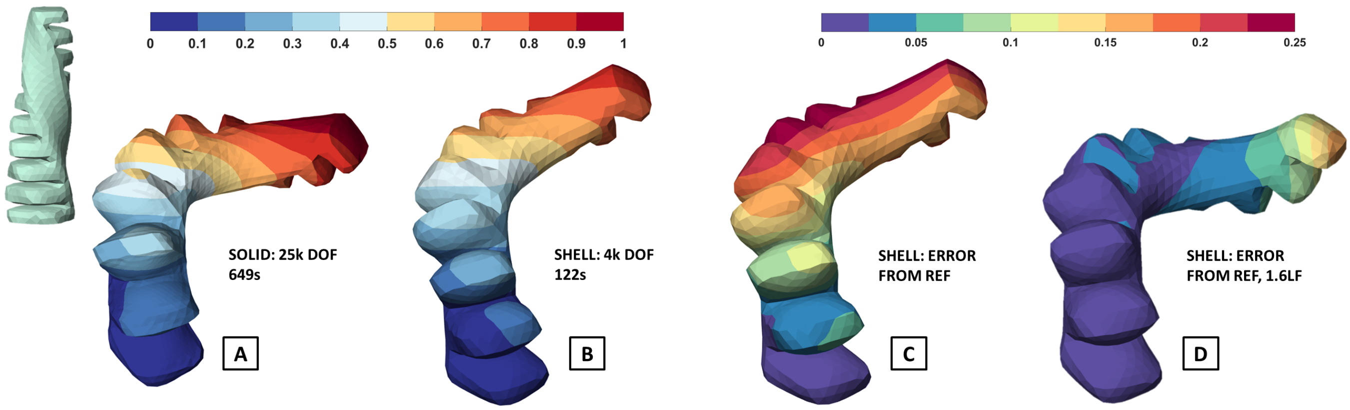

Fig. 5. FE simulation of a soft actuator designed using SoRoForge which displays compound twisting and bending behavior in response to applied internal pressure. A and B give results of simulations executed using tetrahedral volumetric (reference) and triangular shell elements respectively, with contours showing displacement magnitude normalized by actuator length. As expected, simulation using triangular elements exhibits less deformation due to numerical locking. Contours in C and D show local L2 norm of difference in position predicted by shell simulation relative to reference volumetric simulation, using equivalent pressure load to reference simulation and load scaled by 1.6x, respectively. Applying a scaled load reduces normalized error in shell simulations, in this case bringing error to below 10% of actuator length for the majority of the design.

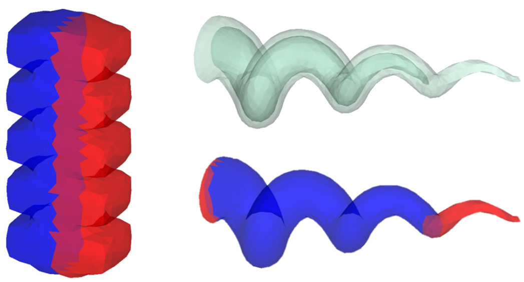

Fig. 6. Adjustment of boundary conditions to account for multi-chambered actuators and wall thickness effects.

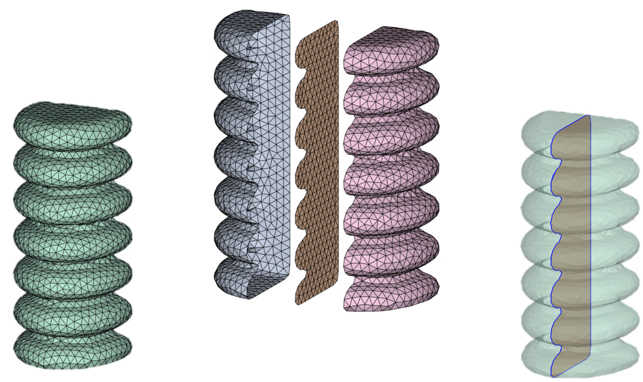

Fig. 7. Automated augmentation of a shell finite element mesh to include an internal membrane. From left: original triangulation derived from implicit geometry function, splitting the geometry along the medial plane and generating a membrane; merging the three resulting meshes into a final FE mesh.

within 10% of the total actuator length for the majority of the design. For the purposes of quickly predicting the deformation modes of a candidate actuator design, shell elements strike a favorable balance between simulation speed and accuracy. Our further work exploring the use of shell finite elements in this context demonstrates similar results in terms of accuracy and computational cost [59].

SoRoForge leverages the open-source package Finite Elements for Biomechanics (FEBio), a general and extensible nonlinear FE code with 10 years of development [21] to perform simulation of candidate designs. FEBio was developed specifically for biomechanics research, and specializes in strongly nonlinear continuum mechanics problems involving large deformations, nonlinear material models, and contact boundary conditions. FEBio is also highly scriptable, so SoRoForge automatically builds, dispatches, and interprets simulations in the background, allowing users to execute advanced simulations with the click of one button.

We incorporate the meshing capabilities of the Geometry and Image-Based Bioengineering add-On (GIBBON) [20] to extract high-quality triangular meshes suitable for finite elements from scalar field evaluations of computational networks. Users may edit simulation settings including shell thickness, pressure magnitude, element size, and material Shore A hardness within SoRoForge, allowing rapid exploration of different simulation conditions on a candidate geometry. Triangulation data is written to an input file as the location matrix for a finite element mesh, and additional data is automatically added to this input file, including boundary conditions, material models, solution settings, and output variables to monitor. Many of these setting are user-controllable, including the actuator material model, actuator wall thickness, applied pressure, mesh resolution, and boundary conditions. The completed input file is passed to the FEBio solver through the command line automatically, bypassing the FEBio GUI. All simulations presented in this paper are performed on identical hardware: a fast but affordable desktop PC with an AMD Ryzen 3900x CPU (24 threads, 4.1 GHz) and 64 GB of RAM.

D. Advanced Simulation Features

While the seamless execution of shell simulations and retrieval of results is ideal for rapid design iterations, we recognize that users may occasionally want to conduct more involved evaluation (especially towards the end of a design process). To this end, SoRoForge is capable of automatically generating volumetric tetrahedral meshes and exporting them to the working directory. Users can import these meshes in the FEA tool of their choosing (including FEBio's free GUI interface), and apply more advanced boundary conditions, loading profiles, material models, or contact scenarios. To generate tetrahedral meshes from the computational networks used by SoRoForge, we convert the scalar field representation of the actuator's outer boundary to a signed distance field using GIBBON [20], allowing us to extract a second (interior) surface offset from the outer boundary by a constant distance. We use the open source software *TetGen* [22] to generate tetrahedral meshes between these bounding surfaces.

SoRoForge can also augment the computational mesh prior to simulation in several useful ways. First, users can request that a subset of mesh faces be excluded from the surface set to which pressure is applied. This is a simple means to selectively pressurize different regions of the model, in order to simulate the behavior of actuators with multiple independent pressurization chambers, such as the antagonistic bending actuator in Figure 6, left. In regions of an actuator design where the length scale of geometric features is similar to the wall thickness, the interior and exterior surfaces may diverge significantly in geometry and even topology - note the tip of the spiral actuator in Figure 6, right, which is completely solid. In order to more closely resemble the pressurization of an actual 3D actuator, shell boundary conditions can be adjusted to omit loading on shell faces which are not nearby to the actuator's inner wall surface. This condition can be trivially detected in actuators represented by implicit geometry functions; we simply examine the signed distance field associated with the inner surface of the actuator.

SoRoForge can also add interior surfaces to actuator designs at the user's request - creating a closed non-manifold geometry that accounts for the stiffness of interior membranes. This allows for complete pressurization of the interior surfaces of a multi-chamber soft actuator such as the example in Figure 9. This process is entirely automated and uses a plane to cut an input mesh geometry, creating an additional intermediate surface, and merges the resulting mesh into a final mesh (Figure 7).

E. Automated Fabrication

Although casting remains the dominant fabrication style for soft actuators, researchers have begun to explore additive manufacturing for realizing many soft actuator designs [65]. Fused filament fabrication (FFF) [44], direct silicone extrusion [7], [64], embedded 3D printing [66], inkjet printing [5], [67], [63], and stereolithography [62] have proven effective means of fabricating geometrically complex soft actuators. Advantages of additive manufacturing include relaxing the fabrication constraints inherent in traditional casting techniques [42], rapid and consistent reproduction of existing actuator designs [5], enabling multi-material deposition with high geometric complexity [42], and rapid actuator customization. SoRoForge allows users to export zero-thickness surface triangulations of candidate designs, which can be trivially printed via commodity FFF workflows using zero infill and a constant wall thickness to produce hollow, airtight actuators (Figure 9). Additionally, as noted by [24], there is a natural compatibility between the implicit representations of geometry used in SoRoForge and additive manufacturing, both of which operate on volumetric representations of a part.

TABLE IV
3^{rd}-ORDER OGDEN MATERIAL MODEL

μ_1	0.509	α_1	3.03
μ_2	-20.5	α_2	0.314
μ_3	25.3	α_3	0.175

III. RESULTS: ACTUATOR DESIGNS

Using the workflow described in the previous section, we recreate several notable soft actuator designs from the literature and present them in Figure 8, along with the computational network representation and the simulated response to applied pressure. These designs were recreated and simulated in SoRoForge over a period of several hours by a student user, highlighting the efficiency of the design workflow presented here. Examples include a canonical *PneuNet* bending actuator [39] (A), an antagonistic two-chamber actuator capable of bending in two directions [62] (B), a tentacle design capable of multiaxis bending and extension [31] (C), a low-displacement, high force linear bellows actuator [63] (D), a four-channel tentacle actuator capable of multi-directional motion [64] (E), and a triangle-wave actuator that exhibits bistable "snapping" extension behavior under positive pressure [23] (F).

We note that the implicit geometry representations of these designs are remarkably compact, a sign that the activation functions and computational network structure implemented here are well suited to efficient encoding of soft actuator geometry. We observe that using SoRoForge makes designing with implicit geometry functions intuitive, and the ability to change network structure or weights and receive immediate visual feedback accelerates the design iteration process. The designs presented in this work, along with dozens of others, are included in the default SoRoForge library so that users can access them and load them into SoRoForge to make their own alterations.

Empirical testing confirms that the shell-based FE simulation approach adopted in this work is useful in predicting the behavior of fabricated actuators. We fabricated antagonistic bending actuators designed in our workflow by specifying a 1.6*mm* wall thickness, extending inward from the surface defined by Eq. 1 (modified by automatically adding an interior membrane in the medial plane). Actuators were fabricated from the thermoplastic polyurethane (TPU) material *NinjaFlex* (©NinjaTek), which is tough, compliant (Shore 85A hardness), and can be readily used by many commercial 3D printers.

We subjected actuators to a quasistatic 40 kPa internal pressure load in both simulation and experiment, which is consistent with pressure loading used in published soft actuators [42]. The simulation was conducted on a 2*k* DOF shell-based finite element mesh, using a third order Ogden model derived uniaxial tension test data, given in Table IV. Material models for many common FFF filament options

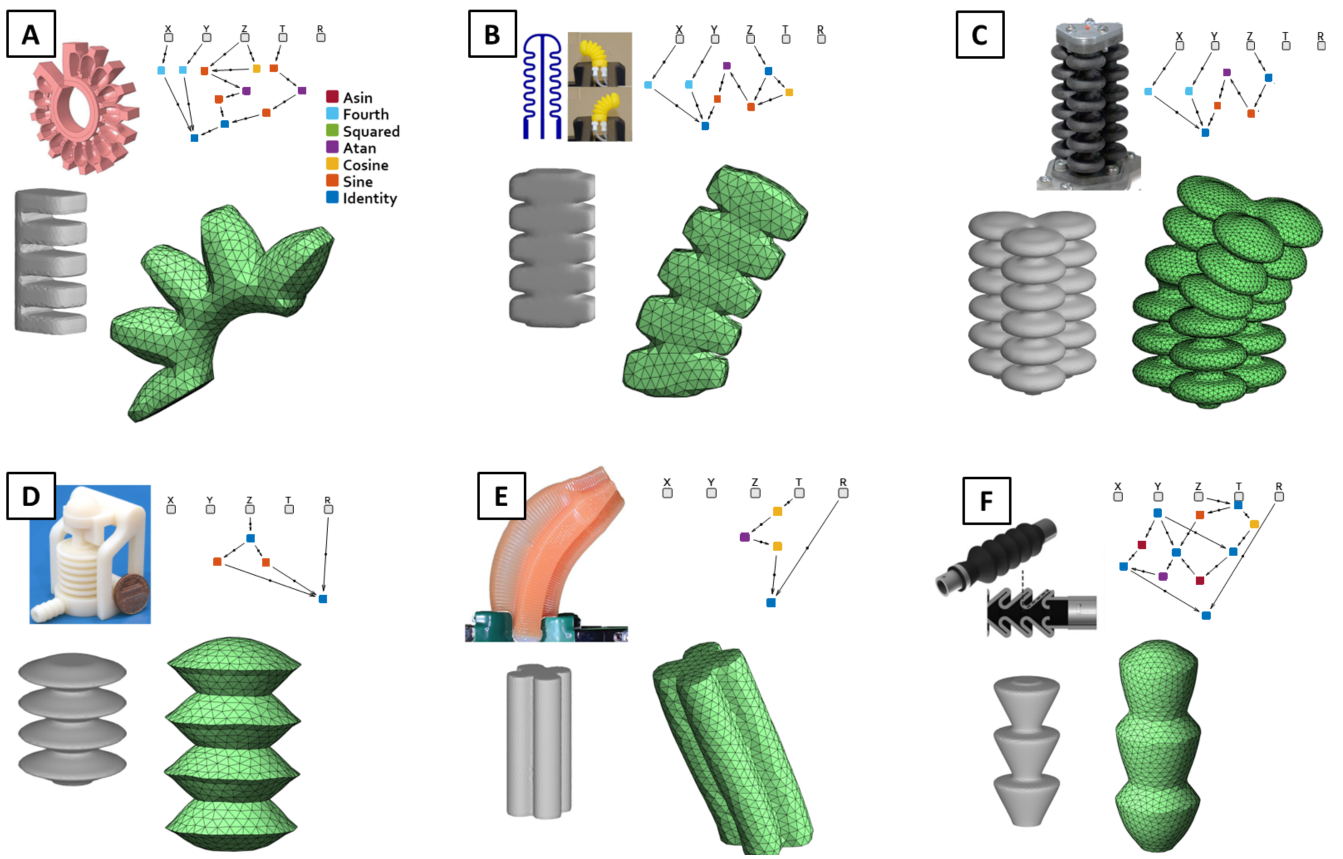

Fig. 8. Six notable actuator morphologies from literature manually recreated using SoRoForge, with computational network representation and deformed actuator geometry under pneumatic load shown. Examples include a canonical *PneuNet* bending actuator [39] (A), an antagonistic two-chamber actuator capable of bending in two directions [62] (B), a tentacle design capable of multiaxis bending and extension [31] (C), a low-displacement, high force linear bellows actuator [63] (D), a four-channel tentacle actuator capable of multidirectional motion [64] (E), and a triangle-wave actuator that exhibits bistable "snapping" extension behavior under positive pressure [23] (F).

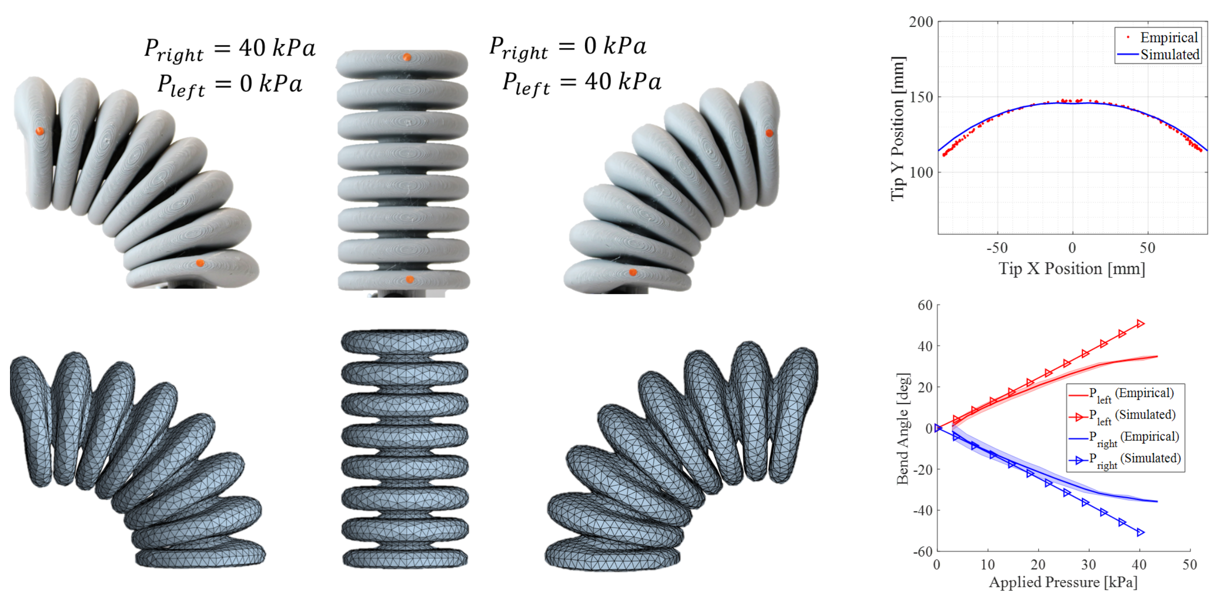

Fig. 9. Simulated and empirical results for antagonistic bending actuator deformation under pressure loading. Simulation captures actuator behavior well when comparing tip displacement, though shell finite elements overestimate the stiffness of the structure. Computations executed in 42s, error bars indicate 2σ variation in measured angle, $n = 8$ trials.

are readily available in SoRoForge simply by specifying the shore-hardness of the desired material.

We track displacement at the tip of the actuator and plot the angle of a line connecting each end of the actuator against the applied pneumatic load (see Figure 9). Simulations predict the shape of the deformed actuator relatively well, though as expected they overestimate the stiffness of the structure - particularly evident at low pressure loads. We reiterate that this is an acceptable trade-off for the speedup realized, and speed is paramount during the design process, when many candidate designs must be evaluated.

IV. Conclusion and Further Research

We present a streamlined, integrated tool for design, simulation, and fabrication of multimaterial soft robotic actuators. By compromising slightly on the accuracy of simulation results and the comforting habits of interactive CAD, our process makes gains in ease of use, speed, and inter-connectivity between different stages of the soft robotics design and fabrication process. In particular, we extend our previous work in [4] to present for the first time a soft actuator design tool with:

- complete integration of simulation execution and retrieval, with control over additional parameters, live monitoring of simulation progress, and in-app results viewer
- improved triangular mesh generation using open source meshing software
- expanded palette of elementary mathematical functions for construction of computational networks
- improved live updates to 3D visualization so users can see effects of small changes
- automatic conversion of computational network representation to symbolic form for increased usability
- automatic generation of internal non-manifold features required for the operation of antagonistic actuators
- deployable instance of explorer app on a virtual machine for easier distribution

We hope to see extensions made to the open-source software tool presented here which improve the generality of the tool while maintaining the "seamless" nature of the workflow. Our shell-based finite element simulation approach could be augmented to include anisotropic material models appropriate for simulating fiber/elastomer composites common in soft actuators [48], [68] (though not usually 3D printed). This is a potentially exciting opportunity, and would dovetail with recent research in deriving flat patterns for 3D inflatable objects [69], [70], but would require additional SoRoForge development to specify local fiber orientation for anisotropic materials. We note that a vector-valued function evaluated at the surface of a SoRoForge actuator could readily achieve this end and could easily be represented by an additional computational network in the same style as the networks used to specify actuator geometry described in this paper.

In the work presented here, print "slicing" and design checking is performed offline by human designers - our tool could be improved to automatically evaluate designs for printability and return feedback to the designer. These checks could include detection and automatic support generation for overhangs (for example as implemented in [71] and [72]), multi-material incompatibility (e.g. [63]), or minimum wall thickness for airtightness (as reported by [42], [73], and [44] and others). For the work presented here, we avoid the use of support material entirely, fabricating the antagonistic finger in Figure 9 using the natural "bridging" capabilities of hobby-grade fused filament fabrication printers. Clearly, fabrication limitations and associated design requirements are machine-dependent, and full compatibility with users' chosen fabrication modality requires development of machine-aware design checking. Integration of the slicing process used to generate extruder toolpaths for fused filament fabrication printing into SoRoForge could enable online fabrication-aware design.

The addition of a pneumatic connection(s) to a soft actuator design is left to the user as a manual post-process. No consensus on pneumatic connection for soft actuators has emerged in literature; connector design typically is adapted to the pneumatic power source at hand. For example, [73] employs printed quick-connect adapters, while many others simply design a hole correctly sized to a standard tubing size [23], [42], [74], [75], still others use independently fabricated connectors [45], and [39] simply puncture actuators with a needle to form a pneumatic connection. In future work, we envision including a menu of pneumatic connectors for users to choose from, as well as automatic merging of connector geometry using implicit geometry boolean operations described in Section II

SoRoForge enables non-experts to rapidly design and improve soft robotic actuators, and integrates with existing additive manufacturing and emerging automated design tools. By deploying our tool as open-source software and publicizing it on a dedicated website which stores and displays SoRoForge designs, we expect that many users will interact with this workflow. Users will upload, download, and remix others' soft robot designs, and we hope this accelerates progress in the soft robotics field.

We intend to analyze this rich dataset assembled by many human designers, and extract insights about implicit geometry modeling of high-performing soft actuators. The computational network representations of actuator geometry that we have chosen for our workflow are ideally suited for this analysis, because we can borrow heavily from extensive theoretical and practical work in graph theory. The insights gained from analysis of community-sourced soft robot designs will be critical to realizing workflows for automated design and fabrication of soft actuators. With the design insights garnered from community use of the tool presented here, we plan to extend our existing work in automated synthesis of soft actuators [16].

This work was supported by MACLab startup funds and is publicly accessible at https://github.com/MacCurdyLab/Soft-Actuator-Synthesis and www.SoRoForge.com.

References

[1] D. P. Holland, E. J. Park, P. Polygerinos, G. J. Bennett, and C. J. Walsh, "The soft robotics toolkit: Shared resources for research and design," *Soft Robot.*, vol. 1, no. 3, pp. 224–230, Sep. 2014.

[2] E. W. Hawkes, C. Majidi, and M. T. Tolley, "Hard questions for soft robotics," *Sci. Robot.*, vol. 6, no. 53, Apr. 2021, Art. no. eabg6049.

[3] O. Sigmund and K. Maute, "Topology optimization approaches: A comparative review," *Struct. Multidisciplinary Optim.*, vol. 48, no. 6, pp. 1031–1055, Dec. 2013.

[4] L. Smith, T. Hainsworth, Z. Jordan, X. Bell, and R. MacCurdy, "A seamless workflow for design and fabrication of multimaterial pneumatic soft actuators," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 718–723.

[5] C. Du Pasquier, T. Chen, S. Tibbits, and K. Shea, "Design and computational modeling of a 3D printed pneumatic toolkit for soft robotics," *Soft Robot.*, vol. 6, no. 5, pp. 657–663, Oct. 2019.

[6] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015.

[7] O. Byrne et al., "Additive manufacture of composite soft pneumatic actuators," *Soft Robot.*, vol. 5, no. 6, pp. 726–736, Dec. 2018.

[8] P. Moseley, J. M. Florez, H. A. Sonar, G. Agarwal, W. Curtin, and J. Paik, "Modeling, design, and development of soft pneumatic actuators with finite element method," *Adv. Eng. Mater.*, vol. 18, no. 6, pp. 978–988, Jun. 2016.

[9] B. Caasenbrood. (2020). *Sorotoki—A Soft Robotics Toolkit for MATLAB*. [Online]. Available: https://github.com/BJCaasenbrood/SorotokiCode

[10] S. Sahabi and J. Bender, "SoRoCAD 2.0: Extending a CAD tool for soft robotics," Bachelor's Thesis, Dept. Comput. Sci., Media Comput. Group, RWTH Aachen Univ., Aachen, Germany, 2021.

[11] Y. Yao, L. He, and P. Maiolino, "A simulation-based toolbox to expedite the digital design of bellow soft pneumatic actuators," in *Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2022, pp. 29–34.

[12] A. Brocker, J. Strüver, S. Voelker, and J. Borchers, "Sorocad: A design tool for the building blocks of pneumatic soft robotics," in *Proc. CHI Conf. Hum. Factors Comput. Syst. Extended Abstr.* New York, NY, USA: Association for Computing Machinery, 2022, pp. 1–7.

[13] F. Faure et al., "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery* (Studies in Mechanobiology, Tissue Engineering and Biomaterials), vol. 11, Y. Payan, Ed. Springer, 2012, pp. 283–321.

[14] J. Strüver, "SoRoCAD: Enabling users to design custom soft robotics," M.S. thesis, Dept. Comput. Sci., Media Comput. Group, RWTH Aachen Univ., Aachen, Germany, 2020.

[15] B. Caasenbrood, A. Pogromsky, and H. Nijmeijer, "A computational design framework for pressure-driven soft robots through nonlinear topology optimization," in *Proc. 3rd IEEE Int. Conf. Soft Robot. (RoboSoft)*, Jul. 2020, pp. 633–638.

[16] L. Smith, T. Hainsworth, J. Haimes, and R. MacCurdy, "Automated synthesis of bending pneumatic soft actuators," in *Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2022, pp. 358–363.

[17] B. Caasenbrood. (2019). *Efficient Soft Robotic Models for (Real-Time) Control Applications*. [Online]. Available: https://github.com/BJCaasenbrood/soft-robot-model-MATLAB

[18] B. J. Caasenbrood, F. E. van Beek, H. K. Chu, and I. A. Kuling, "A desktop-sized platform for real-time control applications of pneumatic soft robots," in *Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2022, pp. 217–223.

[19] S. M. H. Sadati, S. E. Naghibi, L. Da Cruz, and C. Bergeles, "Reduced-order modeling and model order reduction for soft robots," 2020.

[20] K. M. Moerman, "GIBBON: The geometry and image-based bioengineering add-on," *J. Open Source Softw.*, vol. 3, no. 22, p. 506, Feb. 2018.

[21] S. A. Maas, B. J. Ellis, G. A. Ateshian, and J. A. Weiss, "FEBio: Finite elements for biomechanics," *J. Biomech. Eng.*, vol. 134, no. 1, p. 11005, Jan. 2012.

[22] H. Si, "TetGen, a delaunay-based quality tetrahedral mesh generator," *ACM Trans. Math. Softw.*, vol. 41, no. 2, pp. 1–36, Feb. 2015.

[23] B. Sparrman et al., "Printed silicone pneumatic actuators for soft robotics," *Additive Manuf.*, vol. 40, Apr. 2021, Art. no. 101860.

[24] Q. Li, Q. Hong, Q. Qi, X. Ma, X. Han, and J. Tian, "Towards additive manufacturing oriented geometric modeling using implicit functions," *Vis. Comput. Ind., Biomed., Art*, vol. 1, no. 1, p. 9, Dec. 2018.

[25] A. Bernhardt, L. Barthe, M.-P. Cani, and B. Wyvill, "Implicit blending revisited," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 367–375, May 2010.

[26] G. Allen, "nTopology modeling technology," nTopology, New York, NY, USA, White Paper, 2020.

[27] J. Hiller and H. Lipson, "Automatic design and manufacture of soft robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 457–466, Apr. 2012.

[28] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson, "Unshackling evolution," *ACM Sigevolution*, vol. 7, no. 1, pp. 11–23, Aug. 2014.

[29] R. Kicinger, T. Arciszewski, and K. D. Jong, "Evolutionary computation and structural design: A survey of the state-of-the-art," *Comput. Struct.*, vol. 83, nos. 23–24, pp. 1943–1978, Sep. 2005.

[30] M. Pozzi et al., "Efficient FEM-based simulation of soft robots modeled as kinematic chains," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4206–4213.

[31] G. Fang, C.-D. Matte, R. B. N. Scharff, T.-H. Kwok, and C. C. L. Wang, "Kinematics of soft robots by geometric computing," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1272–1286, Aug. 2020.

[32] Y. Hu et al., "ChainQueen: A real-time differentiable physical simulator for soft robotics," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, 2018, pp. 6265–6271.

[33] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning mesh-based simulation with graph networks," 2020, *arXiv:2010.03409*.

[34] R. Luo, S. Member, T. Shao, and H. Wang, "NNWarp: Neural network-based nonlinear deformation," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 4, pp. 1745–1759, Apr. 2020.

[35] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1565–1576, Dec. 2018.

[36] Y. Tang et al., "Leveraging elastic instabilities for amplified performance: Spine-inspired high-speed and high-force soft robots," *Sci. Adv.*, vol. 6, no. 19, May 2020, Art. no. eaaz6912.

[37] G. R. Joldes, A. Wittek, and K. Miller, "Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation," *Med. Image Anal.*, vol. 13, no. 6, pp. 912–919, Dec. 2009.

[38] J. Till and D. C. Rucker, "Elastic stability of cosserat rods and parallel continuum robots," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 718–733, Jun. 2017.

[39] B. Mosadegh et al., "Pneumatic networks for soft robotics that actuate rapidly," *Adv. Funct. Mater.*, vol. 24, no. 15, pp. 2163–2170, Apr. 2014.

[40] L. Ding et al., "Design of soft multi-material pneumatic actuators based on principal strain field," *Mater. Design*, vol. 182, Nov. 2019, Art. no. 108000.

[41] M. Schaffner, J. A. Faber, L. Pianegonda, P. A. Rühs, F. Coulter, and A. R. Studart, "3D printing of robotic soft actuators with programmable bioinspired architectures," *Nature Commun.*, vol. 9, no. 1, p. 878, Feb. 2018.

[42] T. Hainsworth, L. Smith, S. Alexander, and R. MacCurdy, "A fabrication free, 3D printed, multi-material, self-sensing soft actuator," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4118–4125, Jul. 2020.

[43] W. Xiao, D. Hu, W. Chen, G. Yang, and X. Han, "A new type of soft pneumatic torsional actuator with helical chambers for flexible machines," *J. Mech. Robot.*, vol. 13, no. 1, Feb. 2021, Art. no. 011003.

[44] H. K. Yap, H. Y. Ng, and C.-H. Yeow, "High-force soft printable pneumatics for soft robotic applications," *Soft Robot.*, vol. 3, no. 3, pp. 144–158, Sep. 2016.

[45] G. Zhong, W. Dou, X. Zhang, and H. Yi, "Bending analysis and contact force modeling of soft pneumatic actuators with pleated structures," *Int. J. Mech. Sci.*, vol. 193, Mar. 2021, Art. no. 106150.

[46] S. Wakimoto, K. Suzumori, and K. Ogura, "Miniature pneumatic curling rubber actuator generating bidirectional motion with one air-supply tube," *Adv. Robot.*, vol. 25, nos. 9–10, pp. 1311–1330, Jan. 2011.

[47] K. Wandke, "MOOSE-based finite element hyperelastic modeling for soft robot simulations," *IEEE Access*, vol. 9, pp. 139627–139635, 2021.

[48] P. H. Nguyen and W. Zhang, "Design and computational modeling of fabric soft pneumatic actuators for wearable assistive devices," *Sci. Rep.*, vol. 10, no. 1, p. 9638, Jun. 2020.

[49] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[50] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 2, pp. 299–307, Jun. 2003.

[51] C. B. Jones, "An approach to splitting atoms safely," *Electron. Notes Theor. Comput. Sci.*, vol. 155, pp. 43–60, May 2006.

[52] G. Gu, D. Wang, L. Ge, and X. Zhu, "Analytical modeling and design of generalized pneu-net soft actuators with three-dimensional deformations," *Soft Robot.*, vol. 8, no. 4, pp. 462–477, Aug. 2021.

[53] K. M. de Payrebrune and O. M. O'Reilly, "On constitutive relations for a rod-based model of a pneu-net bending actuator," *Extreme Mech. Lett.*, vol. 8, pp. 38–46, Sep. 2016.

[54] X. Liu, Y. Zhao, D. Geng, S. Chen, X. Tan, and C. Cao, "Soft humanoid hands with large grasping force enabled by flexible hybrid pneumatic actuators," *Soft Robot.*, vol. 8, no. 2, pp. 175–185, Apr. 2021.

[55] X. Zhou, C. Majidi, and O. M. O'Reilly, "Soft hands: An analysis of some gripping mechanisms in soft robot design," *Int. J. Solids Struct.*, vols. 64–65, pp. 155–165, Jul. 2015.

[56] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, May 2019.

[57] I. Hussain, Z. Iqbal, M. Malvezzi, L. Seneviratne, D. Gan, and D. Prattichizzo, "Modeling and prototyping of a soft prosthetic hand exploiting joint compliance and modularity," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 65–70.

[58] G. Urbain, J. Degrave, B. Carette, J. Dambre, and F. Wyffels, "Morphological properties of mass–spring networks for optimal locomotion learning," *Frontiers Neurorobotics*, vol. 11, p. 16, Mar. 2017.

[59] L. Smith, J. Haimes, and R. MacCurdy, "Stretching the boundary: Shell finite elements for pneumatic soft actuators," in *Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2022, pp. 403–408.

[60] M. Bischoff, K.-U. Bletzinger, W. A. Wall, and E. Ramm, "Models and finite elements for thin-walled structures," in *Encyclopedia of Computational Mechanics: Solids, Structures and Coupled Problems*, vol. 2, E. Stein, R. de Borst, and T. J. R. Hughes, Eds. Hoboken, NJ, USA: Wiley, Nov. 2004, ch. 3, pp. 1–25.

[61] P.-S. Lee, H.-C. Noh, and K.-J. Bathe, "Insight into 3-node triangular shell finite elements: The effects of element isotropy and mesh patterns," *Comput. Struct.*, vol. 85, nos. 7–8, pp. 404–418, Apr. 2007.

[62] B. N. Peele, T. J. Wallin, H. Zhao, and R. F. Shepherd, "3D printing antagonistic systems of artificial muscle using projection stereolithography," *Bioinspiration Biomimetics*, vol. 10, no. 5, Sep. 2015, Art. no. 055003.

[63] R. MacCurdy, R. Katzschmann, Y. Kim, and D. Rus, "Printable hydraulics: A method for fabricating robots by 3D co-printing solids and liquids," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3878–3885.

[64] O. D. Yirmibesoglu et al., "Direct 3D printing of silicone elastomer soft robots and their performance comparison with molded counterparts," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 295–302.

[65] A. Zolfagharian, A. Z. Kouzani, S. Y. Khoo, A. A. A. Moghadam, I. Gibson, and A. Kaynak, "Evolution of 3D printed soft actuators," *Sens. Actuators A, Phys.*, vol. 250, pp. 258–272, Oct. 2016.

[66] R. L. Truby et al., "Soft somatosensitive actuators via embedded 3D printing," *Adv. Mater.*, vol. 30, no. 15, Apr. 2018, Art. no. 1706383.

[67] H. Zhang, M. Y. Wang, F. Chen, Y. Wang, A. S. Kumar, and J. Y. H. Fuh, "Design and development of a soft gripper with topology optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6239–6244.

[68] F. Connolly, D. A. Wagner, C. J. Walsh, and K. Bertoldi, "Sew-free anisotropic textile composites for rapid design and manufacturing of soft wearable robots," *Extreme Mech. Lett.*, vol. 27, pp. 52–58, Feb. 2019.

[69] J. Panetta, F. Isvoranu, T. Chen, E. Siéfert, B. Roman, and M. Pauly, "Computational inverse design of surface-based inflatables," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–14, Aug. 2021.

[70] A. E. Forte et al., "Inverse design of inflatable soft membranes through machine learning," *Adv. Funct. Mater.*, vol. 32, no. 16, Apr. 2022, Art. no. 2111610.

[71] W. Wang et al., "Support-free hollowing," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 10, pp. 2787–2798, Oct. 2018.

[72] J. Wu, C. C. L. Wang, X. Zhang, and R. Westermann, "Self-supporting rhombic infill structures for additive manufacturing," *Comput.-Aided Des.*, vol. 80, pp. 32–42, Nov. 2016.

[73] G. Stano, L. Arleo, and G. Percoco, "Additive manufacturing for soft robotics: Design and fabrication of airtight, monolithic bending PneuNets with embedded air connectors," *Micromachines*, vol. 11, no. 5, p. 485, May 2020.

[74] J. Fras, M. Macias, Y. Noh, and K. Althoefer, "Fluidical bending actuator designed for soft octopus robot tentacle," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 253–257.

[75] T. Kanno and K. Kawashima, "External force estimation of pneumatic soft actuator with built-in displacement sensor," *Sensors Mater.*, vol. 33, no. 2, p. 555, 2021.