Model Reduction in Soft Robotics Using Locally Volume-Preserving Primitives

Yi Xu and Gregory S. Chirikjian, *Fellow, IEEE*

***Abstract*—A new, and extremely efficient, computational modeling paradigm is introduced here for specific finite elasticity problems that arise in the context of soft robotics. Whereas continuum mechanics is a very classical area of study that is broadly applicable throughout engineering, and significant effort has been devoted to the development of intricate constitutive models for finite elasticity, we show that for the most part, the isochoric (locally volume-preserving) constraint dominates behavior in common soft robotics contexts, and this can be built into closed-form kinematic deformation fields before even considering other aspects of constitutive modeling. We therefore focus on developing and applying primitive deformations that each observe this constraint. By composing a wide enough variety of such deformations, many of the most common behaviors observed in soft robots can be replicated. Case studies include isotropic objects subjected to different boundary conditions, a non-isotropic helically-reinforced tube, and a not-purely-kinematic scenario with gravity loading. We show that this method is at least 50 times faster than the ABAQUS implementation of the finite element method (FEM), and has speed comparable with the real-time FEM framework SOFA. Experiments show that both our method and ABAQUS have approximately 10% error relative to experimentally measured displacements, as well as to each other. And our method outperforms SOFA when the deformation is highly nonlinear. Our method provides a real-time alternative to FEM, and captures essential degrees of freedom for potential use in feedback control systems.**

***Index Terms*—Modeling, control, and learning for soft robots, kinematics.**

I. Introduction

CONTINUUM mechanics and finite elasticity are classical fields wherein many research topics are considered solved, and acceptable answers to engineering problems can be provided by using a finite element method (FEM) code. To perform real-time feedback control, it is desirable for the computational model of the physical system to update tens or hundreds of times per second. Whereas such calculations can be distributed over multiple processors to increase speed to enable real-time use, we introduce an alternative that is computationally efficient and fast even on a single processor. Specifically, we propose an alternative modeling paradigm for soft structures, akin to Galerkin methods in linear elasticity. In real life, the incompressibility assumption holds for rubber-like materials. Hence, every differential volume element of the elastic material could be treated as volume-preserving. The deformation can then be modeled by local volume-preserving deformations introduced by Chirikjian [1], where several closed-form deformation primitives were defined. Fig. 1 illustrates several primitives. Our method exhibits the following characteristics:

- The method utilizes deformation primitives that can describe continuous deformation fields without mesh.
- Complex deformation can be constructed through the nonlinear functional composition of primitives. The reasoning is that if each primitive deformation is isochoric, then so too will be the composition.
- The configuration of deformation fields can be adjusted by changing the coefficients in primitives.

Our method based on these primitives cuts down the DOFs, enabling rapid simulation. More details about the method will follow, but first some background is provided.

Soft robots are fabricated from highly pliable materials (e.g. silicone, textiles, or cables) [2], which allow the robots to reshape continuously and softly with simple actuation. In recent years, the design and modeling of soft robots has emerged as one of the most significant evolutions in the field of robotics. The development of simulation and analysis tools has become one

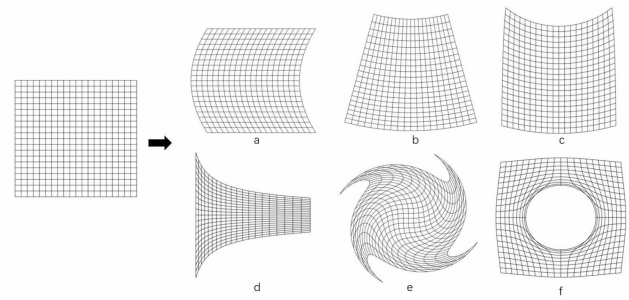

Fig. 1. Locally volume preserving deformation primitives: a. shear; b. variable offset bending; c. offset shear bending; d. elongation and contraction; e. vortex; f. source. [1] The mesh shown in the figure is solely included to aid in visualizing the deformation. Our method is actually meshless in nature.

Manuscript received 27 February 2023; accepted 5 July 2023. Date of publication 31 July 2023; date of current version 8 August 2023. This letter was recommended for publication by Associate Editor E. Falotico and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported by NUS Startup under Grants A-0009059-02-00, A-0009059-03-00, in part by CDE Board account under Grant E-465-00-0009-01, in part by MOE Tier 2 under Grant REBOT A-8000424-00-00, and in part by the National Research Foundation, Singapore, through Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN), under sub Award A-0009428-08-00. *(Corresponding author: Gregory S. Chirikjian.)*

The authors are with the Department of Mechanical Engineering, National University of Singapore, Singapore 117575 (e-mail: x.yi@u.nus.edu; mpegre@nus.edu.sg).

Digital Object Identifier 10.1109/LRA.2023.3300226

of the major topics in this research field [3], [4], [5]. Several current modeling methods are reviewed here.

The majority of the current models used in soft robotics are mechanics-based such as FEM, which plays a significant role in soft structure research [6], [7]. Open-source frameworks are also developed for modeling soft robots [8], [9]. FEM handles complex simulations, such as structures with multiple of materials or structures interacting with the surroundings. FEM specializes in representing soft robot configurations precisely since it follows closely to the principle laws, while providing researchers sufficient degrees of freedom (DOFs). However, it involves extensive computational processes due to its complexity, which requires users to construct mesh modeling of objects. Significant research efforts have been put on developing real-time FEM modeling [10]. Implementing jobs on Graphics Processing Units (GPU) instead of Central Processing Unit (CPU) can significantly improve the speed as GPU provides massive computation power [11]. Researchers also developed optimized solvers that balances the accuracy and speed according to the requirement [12]. Largilliere and Duriez et al. [13], [14], [15] presented a FEM model with a frequency response of 600 Hz, and developed a plugin for the SOFA Framework using this method. By combining reduced-order algorithms and FEM, real-time simulation can also be achieved [16]. Real-time FEM also requires a mesh to be generated, the interpolation between nodes usually reduce the accuracy.

For systems with limited complexity, researchers also developed their own models to capture the motion of soft actuators. Polygerinos et al. [17] modeled the principle of fiber-reinforced bending actuators with a quasi-static model; Jones et al. [18] used Cosserat rod theory to capture the configuration of a continuum robot; Della Santina et al. [19] modeled a planar bending actuator based on the piecewise constant curvature hypothesis. These methods have high accuracy, but also require complex computation.

Methods without constitutive laws also have been proposed. Learning based algorithms [20] analyze data obtained from physical experiments to generate the optimal deformation parameters. Neural network approaches [21], [22] formulate the discrete state-space representation of the system, which enabled position control within a small deformation. Deep learning [23] and regression analysis [24] also derive the relationship between bending angle and input pressure. Learning based methods skip the tedious calculation of energy minimization, but they heavily rely on a large number of input data and repeated training processes.

II. Methods

The process in our modeling approach is demonstrated in Fig. 2. The details are discussed in this section.

A. Locally Volume Preserving Deformation Primitives

All the modeling works are based on locally volume preserving deformations. A deformation $\mathbf{f}$ maps a set of referential

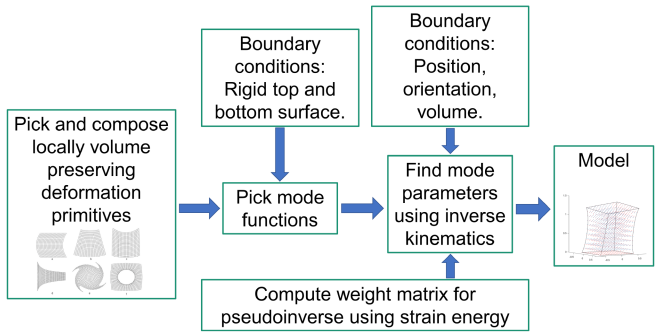

Fig. 2. Modeling process.

Cartesian coordinates $\mathbf{x} = [x_1, x_2, x_3]^T$ into a new set of coordinates $\mathbf{f} = \mathbf{f}(\mathbf{x}) = [f_1, f_2, f_3]^T$. T denotes the transpose of a vector or matrix.

For locally volume preserving deformation,

$$\det(\nabla_{\mathbf{x}}\mathbf{f}) = 1 \tag{1}$$

Given two locally volume preserving deformation $\mathbf{f}_1(\mathbf{x})$ and $\mathbf{f}_2(\mathbf{x})$, the determinant of gradient of composition $\mathbf{f}_1(\mathbf{f}_2(\mathbf{x}))$ is:

$$\det(\nabla_{\mathbf{x}}\mathbf{f}_1(\mathbf{f}_2(\mathbf{x}))) = \det(\nabla_{\mathbf{y}}\mathbf{f}_1(\mathbf{y}))\det(\nabla_{\mathbf{x}}\mathbf{f}_2(\mathbf{x})) = 1 \cdot 1 = 1 \tag{2}$$

Therefore, the compositions of locally volume preserving deformations are also locally volume preserving.

Soft materials are considered incompressible, hence (1) is usually considered as a constraint when analyzing the deflections of soft materials. The advantage of using locally volume preserve deformation to model soft robots is that the constraint of (1) is implicitly included in the functions, so there is no need to apply this constraint again.

Chirikjian [1] introduced nonlinear locally volume-preserving primitives that serve (1) as a mechanical design tool, which are reviewed and extended here. The primitives used in this letter are introduced below:

a) Stretching and compression: When stretch or compress a block in the z direction, the cube will elongate or shorten in the z direction, and shrink or expand in the x-y plane. If the material is homogeneous, the deformation can be modeled using the nonuniform elongation primitive, $\mathbf{e}(\mathbf{x})$.

$$\mathbf{e}(\mathbf{x}) = \begin{pmatrix} \frac{1}{\sqrt{m_e'(x_3)}} x_1 \\ \frac{1}{\sqrt{m_e'(x_3)}} x_2 \\ m_e(x_3) \end{pmatrix} \tag{3}$$

$m_e'(x_3)$ can be expressed as a weighted sum of the mode functions and reflects the elongation rate at the plane $z = x_3$.

b) Twist: In this primitive, a 3D block is considered as a stack of parallel 2D planes. As the area of each plane is unchanged, the volume of the block is thus preserved after the deformation. Assuming each plane rotates about the x_3 axis by angle $m_\theta(x_3)$,

the deformation can be written as:

$$\mathbf{t}(\mathbf{x}) = \begin{pmatrix} x_1 \cos m_\theta(x_3) - x_2 \sin m_\theta(x_3) \\ x_1 \sin m_\theta(x_3) + x_2 \cos m_\theta(x_3) \\ x_3 \end{pmatrix} \tag{4}$$

$m_\theta(x_3)$ determines the rotation angle at each horizontal plane and can be chosen to be a weighted sum of modes.

c) Shear: A shear deformation $\mathbf{s}(\mathbf{x})$ is defined as below, horizontal planes move along the x_1 and x_2 axis while no deformation exists in the x_3 direction:

$$\mathbf{s}(\mathbf{x}) = \begin{pmatrix} x_1 + m_{s1}(x_3) \\ x_2 + m_{s2}(x_3) \\ x_3 \end{pmatrix} \tag{5}$$

$m_{s1}(x_3), m_{s2}(x_3)$ are the scalar functions that represent the shearing distance in x_1, x_2 direction, respectively. They can be described using modes.

d) Source: The source deformation $\mathbf{c}(\mathbf{x})$ can simulate cavity generation in material. This deformation can model the bulge of chambers made of soft materials after inflation.

$$\mathbf{c}(\mathbf{x}) = \begin{pmatrix} (m_c(x_3) + x_1^2 + x_2^2)^{\frac{1}{2}} \frac{x_1}{(x_1^2+x_2^2)^{\frac{1}{2}}} \\ (m_c(x_3) + x_1^2 + x_2^2)^{\frac{1}{2}} \frac{x_2}{(x_1^2+x_2^2)^{\frac{1}{2}}} \\ x_3 \end{pmatrix} \tag{6}$$

$m_c(x_3)$ is the function expandable in modes. It reflects the radial change of the void within the material.

e) 2D Bend: Bending deformation is defined based on the backbone curve, which captures the configuration of a continuum.

$$\mathbf{b}_2(\mathbf{x}) = \mathbf{a}(x_3) + \frac{1 - \sqrt{1 - 2m_\kappa(x_3)x_2}}{m_\kappa(x_3)}\mathbf{n}(x_3) \tag{7}$$

$\mathbf{a}(x_3)$ is the 2D backbone curve, while $m_\kappa(x_3)$ is the curvature of the backbone curve parameterized by its arclength x_3. The function in front of $\mathbf{n}(x_3)$ is the offset distance of the offset curves that guarentees (1). In order to avoid singularities, the constraint $\frac{1}{2} > m_\kappa x_2$ is applied.

Each primitive involves basic scalar functions that define the shape of the deformations (e.g. area of void in source deformation, bending angle in bending deformation, etc.). The basic scalar functions can be expanded in a series of weighted mode functions in the form as in [27]:

$$m(x) = \sum_{i=1}^{M} p_i \phi_i(x), \tag{8}$$

Here $\phi_i(x)$ are the mode functions, p_i are the weight parameters, and M is the number of modes. For example:

$$m(x) = p_1 \cdot 1 + p_2 sin(\pi x) + p_3 cos(\pi x), \tag{9}$$

$1, sin(\pi x), cos(\pi x)$ are the mode functions. Once chosen, the functions will be fixed for the model. p_1, p_2, p_3, p_4 are the free parameters to be updated. Typically, mode functions can be chosen as the first few terms in a series of trigonometric functions, which forms a basis with which to approximate any well-behaved function. With sufficient primitive types and modes, an infinite variety of locally volume preserving deformations can be written in closed form by repeated composition.

f) 3D Bend: A 2D bending deformation can be extended to 3D space. Given a 3D bending deformation of this form:

$$\mathbf{b}_3(\mathbf{x}) = \mathbf{a}(x_3) + \nu(x_1, x_2, x_3)\mathbf{n}(x_3) + \beta(x_1, x_2, x_3)\mathbf{b}(x_3) \tag{10}$$

$\mathbf{a}(x_3)$ is the 3D backbone curve, $\mathbf{n}$ is the normal vector and $\mathbf{b}$ is the binormal vector. There are many different choices of ν and β that will result in local volume preservation. A simple one is defined by letting $\beta = x_1$ and $\nu = \frac{1-\sqrt{1-2\kappa x_2}}{\kappa}$, where $\kappa(x_3)$ is the curvature of the backbone curve. Since the Frenet frames twist along the tangent as they traverse the curve, this bending deformation should be composed with a twist deformation along the x_3 axis to form a minimally varying frame. Assume the twist deformation rotates the frame about the x_3 axis by angle $\theta(x_3)$. The new frame has globally minimal twist if [25]:

$$\theta_1(s) = -\int_0^s \tau(\sigma)\, d\sigma, \tag{11}$$

where $\tau(s)$ is the torsion of the curve, and s is the arclength of the curve. In order to preserve the orientation of the bottom plane, let:

$$\theta(x_3) = \theta_1(x_3) + \theta_2, \tag{12}$$

where θ_2 is a constant equal to the angle between the normal vector of the backbone curve at $x_3 = 0$ and the x_2 axis.

B. Boundary Conditions

A common approach to increase the DOFs of soft robots is to connect a series of sections with different functionality. As the connecting structures of the individual sections are usually rigid, it is useful to analyze models with rigid surface boundary conditions applied on both top and bottom.

To incorporate boundary conditions with primitives, we need to arrange the primitives in proper order and select appropriate modes for scalar functions. Commutable primitives such as stretch, twist, shear, and source do not affect the parallelism between horizontal planes, which means the order of compositions is irrelevant. However, bending deformation yields unparallel horizontal planes, which has to be composed last.

After obtaining the appropriate order of primitives, the modes are then constrained based on the boundary conditions applied. Table I formulates the rigid surface boundary conditions applied to the modes introduced above. The mode functions should be tailored to satisfy the boundary conditions automatically. The specific choice of modes is introduced in the later section. Although the boundary condition can also be achieved by adjusting suitable free parameters, choosing the modal basis functions

TABLE I
BOUNDARY CONDITIONS FOR ELONGATION, SHEAR, TWIST, BENDING, AND SOURCE PRIMITIVES

Primitive	Boundary condition	Explanation
Elongation	$\frac{1}{\sqrt{m'_e(0)}} = \frac{1}{\sqrt{m'_e(h)}} = 1$	Limit elongation rate at top and bottom plane to 1
Shear	$m_{s1}(0) = m_{s2}(0) = 0$	Fix the bottom plane
Twist	$m_\theta(0) = 0$	Fix the bottom plane
Bending	$m_\kappa(0) = m_\kappa(h) = 0$	Making the curvature 0 to preserve top and bottom plane
Source	$m_c(0) = m_c(h) = 0$	Set strength of the source to 0 at top and bottom plane

that automatically satisfy the condition reduces the tuning work required for the parameters.

C. Kinematics

After acquiring the composition of primitives, the shape of the model can be controlled by manipulating the mode parameters. If the material is homogeneous and isotropic, when only kinematical boundary conditions are applied, the deformation is not sensitive to material property. Therefore, the configuration of deformed objects can be obtained by minimizing the configuration cost instead of strain energy.

One optimization method we use is the projected gradient method. This method involves moving along the negative gradient of the cost function to minimize cost [26]. The position and orientation of the end effector $\mathbf{x}(\mathbf{p})$ is calculated as a function of mode parameters $\mathbf{p}$. J is the Jacobian matrix of $\mathbf{x}(\mathbf{p})$, similar to the way it is calculated in the method from [27]. Let $Z_k = \mathbb{I} - J^+J$, J^+ is the Jacobian pseudoinverse $J^+ = W^{-1}J^T(JW^{-1}J^T)^{-1}$, where W is the weight matrix. It can be tuned using mechanics to closely describe the deformation, which will be discussed in the next subsection.

The columns of Z_k are the basis for the null space of J. The mode parameters $\mathbf{p}$ are iteratively updated to satisfy the boundary conditions using the following equation.

$$\mathbf{p}_{k+1} = \mathbf{p}_k + J^+(\mathbf{d}_{k+1} - \mathbf{x}(\mathbf{p}_k)) - \alpha Z_k \nabla G(\mathbf{p}_k) \tag{13}$$

where $\mathbf{d}_{k+1}$ and $\mathbf{x}(\mathbf{p}_k)$ are the six dimensional vectors describing the desired and actual position and orientation of the end effector at steps $k+1$ and k, respectively. α is a positive constant smaller than 1. $G(\mathbf{p}_k) = \mathbf{p}_k^T\mathbf{p}_k$ is the configuration cost.

This method can handle simple boundary conditions, but there are better options when 3D rotation is involved. We use the Lie-group-theoretic method by Kim et al. [28] to obtain the minimum energy conformations of objects. This method obtains the Euler–Poincaré equation by applying variational calculus on Lie groups, and solves the equation for boundary conditions. The method does not involve Euler angles, hence avoiding the singularity problem.

Fiber-reinforced situations can be modeled using kinematics as well. The constraint is that the fiber length must be preserved. A good example of such materials is fiber-reinforced pneumatic soft actuators, which consist of a tubular structure made of elastomers and a network of inextensible fibers on the surface for reinforcement. For a twisting actuator with helical fiber $\mathbf{h_f}$:

$$\mathbf{h_f}(l) = \begin{pmatrix} a_f cos\left(\frac{l}{\sqrt{a_f^2+b_f^2}}\right) \\ a_f sin\left(\frac{l}{\sqrt{a_f^2+b_f^2}}\right) \\ b_f \frac{l}{\sqrt{a_f^2+b_f^2}} \end{pmatrix} \tag{14}$$

The fiber is parameterized by arclength l. Upon inflation, the actuator will extend and twist about the x_3 axis to enlarge the chamber volume and preserve the arclength of the fiber. Composition of uniform source, stretch, and twist is used to model the deformation, i.e. $f(x) = t(s(e(x)))$. c, e, θ represents the square of radial change of source, the elongation rate of stretch, and the twisting angle of twist. Parameter e can be expressed as a function of c since the radius of the fiber layer of the actuator r_o is the same before and after the deformation:

$$e = \frac{{r_o}^2 + c}{{r_o}^2} \tag{15}$$

Then from the arclength preserving constraint we can get:

$$\theta = \frac{-a_f + \sqrt{b_f^2 + a_f^2 - b_f^2 e^2}}{a_f b_f}. \tag{16}$$

D. Mechanics

When redundant DOFs are available in the system, using minimization of strain energy to guide the update of parameters can improve the performance of the model. We consider the silicone we used is Mooney-Rivlin solid. Given the deformation gradient $\mathbf{F} = \nabla_{\mathbf{x}}\mathbf{f}$, we can compute the strain tensor $\mathbf{B} = \mathbf{F}^T\mathbf{F}$. The strain energy function of Mooney-Rivlin model is expressed as a function of $\mathbf{B}$ through its principle invarians $I_\mathbf{B} \doteq I_\mathbf{B} \doteq \{I_1(\mathbf{B}), I_2(\mathbf{B}), I_3(\mathbf{B})\}$. For two parameters model, the strain energy density function is given as:

$$\psi = C_{10}(I_1(\mathbf{B}) - 3) + C_{01}(I_2(\mathbf{B}) - 3) + \frac{1}{d}(J-1)K \tag{17}$$

where $J^2 = I_3(\mathbf{B})$. For incompressible materials, $I_3(\mathbf{B}) = 1$. C_{10} and C_{01} are two constant coefficients determined experimentally for any specific material. For Ecoflex 00-30, $C_{10} = 5.6\,kPa$, $C_{01} = 6.3\,kPa$. The strain energy E:

$$E(\mathbf{p}) = \iiint_V \psi \, dx \, dy \, dz \tag{18}$$

Instead of calculate the integral analytically, it is computed numerically by sampling points over the whole body and sum up the strain energy density value times its corresponding small volume. Then use the Newton's method with Lagrange multipliers to solve this constrained optimization problem.

Another strain energy envolved method is to evaluate how much a mode parameter can influence the strain energy, and update the mode parameters in the direction that strain energy is lower. This helps obtaining a configuration with lower strain energy without computing the strain energy in every iteration, which accelerates the computation. The method uses the strain

energy to compute the weight matrix W in the Jacobian pseudoinverse. The pseudoinverse mininizes the quadratic cost c:

$$c = \mathbf{p}^T W \mathbf{p} \tag{19}$$

The following steps can compute the weight matrix W that minimizes the strain energy when deformation is very small. For large deformations, it also helps lower the strain energy of the configuration.

Choose n sets of random small values: $\mathbf{p}_i, \mathbf{p}_2, \ldots, \mathbf{p}_n$. m is the length of $\mathbf{p}$. c_i is the strain energy of the structure when $\mathbf{p}_i$ is used. Then we have:

$$c_i = \sum_{k=1}^{\frac{m(m+1)}{2}} \mathbf{p}_i^T M_k \mathbf{p}_i w_k \tag{20}$$

M_k is a symmetric matrix that has either one entry on the diagonal or two entries on both sides of the diagonal that equal one. The other entries are zero. For example, when $m = 2$, there should be $\frac{m(m+1)}{2} = 3$ different M matrices:

$$M_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, M_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, M_3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{21}$$

(20) can be written as:

$$\sum_{k=1}^{\frac{m(m+1)}{2}} a_{ki} w_k = c_i \tag{22}$$

a_{ki} is the entry ki of matrix A. The above equation can be rewritten as follows:

$$A\mathbf{w} = \mathbf{c} \tag{23}$$

$\mathbf{c}$ contains value from c_1 to c_n. Use unweighted pseudoinverse to solve for $\mathbf{w}$:

$$\mathbf{w} = A^+ \mathbf{c} \tag{24}$$

$\mathbf{w}$ contains all the entries of weight matrix W. Using this weight matrix in the pseudoinverse, the result will converge to a result that has lower total strain energy compared to unweighted ones.

III. Implementation and Comparison With FEM and Physical Experiments

To demonstrate the accuracy of our method, our results are compared with ABAQUS, a software for finite element analysis that deals with nodes. Assuming l nodes are sampled to represent the system. $\mathbf{o}_i(\mathbf{p})$ is the original position of the ith node. $\mathbf{f}_i(\mathbf{p})$ is the position after the deformation computed by primitives, and $\mathbf{a}_i(\mathbf{p})$ is the position computed by ABAQUS. The average error E is calculated as the squared root of the sum of the squared error normalized by the total displacement.

$$E = \left(\frac{\sum_{i=1}^{l} e_i^2}{\sum_{i=1}^{l} d_i^2} \right)^{\frac{1}{2}} = \left(\frac{\sum_{i=1}^{l} \|\mathbf{a}_i(\mathbf{p}) - \mathbf{f}_i(\mathbf{p})\|^2}{\sum_{i=1}^{l} \|\mathbf{a}_i(\mathbf{p}) - \mathbf{o}_i(\mathbf{p})\|^2} \right)^{\frac{1}{2}} \tag{25}$$

d_i represents the displacement of node i. e_i represents the distance difference of node i by two different methods.

Our method is also benchmarked against measurements from the physical experiment. Fig. 3 are the schematic diagram of deformations and corresponding experiment setup. Most of the soft objects in this session are cast using Ecoflex 00-30. An exception is the rod used for testing the model for 3D bending. It is made of Dragonskin 30, which is stiffer than Ecoflex 00-30 and allows us to neglect the impact of gravity during the experiment. All the objects have markers drawn on the surface for coordinate positioning. The configurations of soft objects are captured by Zed 2 stereo cameras calibrated using Zhang's method [29]. The undistorted version of the stereo images can then be obtained and fed into MATLAB computer vision toolbox, which offers a triangulation tool for 3D coordinates positioning.

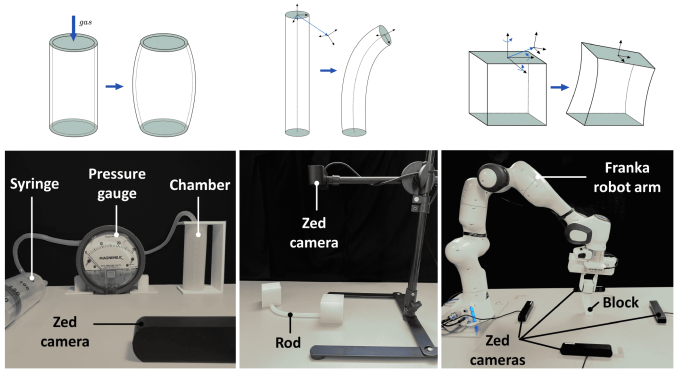

Fig. 3. First row is the schematic diagram of the deformation of chamber, block, and rod. The green surfaces indicate the surfaces are rigid. The second row is the experiment setup. For chamber: a rubber chamber is attached to a rigid frame. A 200 mL syringe is connected to the chamber through a silicone tube. For 2D bending: a rubber rod is placed on a flat horizontal surface. Lubricant is applied to the surface so that the rod can slide on the surface smoothly. The two ends of the rod are attached to two rigid blocks. One block is mounted on the surface and the other can move to get desired position and orientation. A Zed camera is hung over the setup. For 3D bending and block: a rubber block is mounted on the table. A rigid handle is attached to the top surface of the block. It is grasped by a Franka robot arm, which enables precise movement of the surface. Four ZED 2 stereo cameras are located on the side of the block.

A. Modeling With Boundary Conditions: Chamber

Consider a cylindrical chamber with even wall thickness. Upon inflation, the body of the chamber will bulge out. This deformation can be modeled using a single source deformation $\mathbf{c}(\mathbf{x})$. The strength of the source changes along the x_3 axis. Since the chamber has even wall thickness at all heights, it will bulge symmetrically after inflation. Since the boundary condition is simple, (13) is simplified to $\mathbf{p}_{k+1} = \mathbf{p}_k + J^+(d_{k+1} - x(\mathbf{p}_k))$. d_{k+1} and $x(\mathbf{p}_k)$ are the desired and actual volume of the chamber, respectively. Mode functions are chosen as $\sin(\frac{k\pi x_3}{L})$ where k is odd number. Such function can guarantee zero expansion at $x_3 = 0$ and $x_3 = L$ regardless the mode parameters. And the function would be symmetric when $x_3 \in [0, L]$. Hence, the modal expansion is chosen as below:

$$m_c(x_3) = p_1 \sin\left(\frac{\pi x_3}{L}\right) + p_2 \sin\left(\frac{3\pi x_3}{L}\right) + p_3 \sin\left(\frac{5\pi x_3}{L}\right) + p_4 \sin\left(\frac{7\pi x_3}{L}\right) + p_5 \sin\left(\frac{9\pi x_3}{L}\right). \tag{26}$$

During the physical experiment, the chamber can be inflated by pushing the handle of the syringe. Denote the total volume of air inside the chamber, syringe, tube and gauge before the inflation as V_b, atmospheric pressure as P_b. Read the inflation volume V_i and pressure P_a. Assuming the ideal gas law applied, the total volume of the air in the system after inflation becomes $V_a = V_b \frac{P_b}{P_a}$. Then the volume change of the chamber V_d is $V_d = V_i - (V_b - V_a)$. The boundary condition in ABAQUS is the pressure applied to the inner surface of the chamber while the boundary condition in our method is the volume of the chamber.

B. Modeling With Boundary Conditions: Rod

The second case study is a soft rod with a high aspect ratio. In this scenario, bending is the major deformation in the system such that bending primitive $\mathbf{b}_2(\mathbf{x})$ or $\mathbf{b}_3(\mathbf{x})$ dominates the primitives, and other primitives are negligible. Hence, it is sufficient to only model bending in our method. The key to model the deformation of the rod is to capture the "backbone" curve of the rod. It is located at the center of the rod and can be used to describe the shape of the deformed rod.

In the 2D case, the backbone curve can be modeled with curvature scalar function as below.

$$m_\kappa(x_3) = p_1 \sin\left(\frac{\pi x_3}{L}\right) + p_2 \sin\left(\frac{2\pi x_3}{L}\right) + p_3 \sin\left(\frac{3\pi x_3}{L}\right) + p_4 \sin\left(\frac{4\pi x_3}{L}\right) + p_5 \sin\left(\frac{5\pi x_3}{L}\right) + p_6 \sin\left(\frac{6\pi x_3}{L}\right). \quad (27)$$

The chosen mode functions automatically satisfy the condition that the curvature at $x_3 = 0$ and $x_3 = L$ are zero, which preserves the top and bottom planes. In the 3D case, the backbone curve is solved numerically using the method from [28], hence no modes are needed. However, the curvature at $x_3 = 0$ and $x_3 = L$ might not be zero, which leads to the deformation of the plane. For the rod case, this deformation can be neglected since the area of the crosssection is small compared to the height, and the curvature is usually small at these surfaces.

In order to demonstrate the ability of handling external force, we also tested our model on the scenario of an upright rod, where gravity will take effect. The rod leans on a vertical surface to limit the deformation in 2D. The method also uses (13), as in the horizontal rod case. The difference is that the cost function $G(p_k)$ is now the sum of strain energy and gravitational potential energy E_{gp}.

$$E_{gp} = mgh = \rho V_u \sum_{i=1}^{n} \mathbf{b}_2(\mathbf{x}_i) \cdot \mathbf{e}_3 \quad (28)$$

Here ρ is the density of the silicone, n is the number of sample points along the backbone curve, V_u is the unit volume corresponding to each sample section, and $\mathbf{b}_2(\mathbf{x}_i) \cdot \mathbf{e}_3$ is the x_3 coordinate of the ith sample point.

C. Modeling With Boundary Conditions: Block and Film

In this section, a rubber block is modeled to demonstrate the performance of the model with complex boundary conditions. 6 DoFs are identified as the top plane moves to desired positions and orientations, while the bottom plane remains fixed in the process.

TABLE II
MODAL EXPANSIONS FOR MODELING THE RUBBER BLOCK

Deformation	Modal expansion
Twist 1	$ro + twx_3$
Twist 2	$-ro$
Stretch	$(x_3(x_3 - h)st + 1)$
Shear(x_1)	$s_1 x_3$
Shear(x_2)	$s_2 x_3$
Bend	$b_1 \sin(\frac{\pi x_3}{L}) + b_2 \sin(\frac{2\pi x_3}{L}) + b_3 \sin(\frac{3\pi x_3}{L}) + b_4 \sin(\frac{4\pi x_3}{L})$

The twist deformation $\mathbf{t}_1(\mathbf{x})$ gives the top plane freedom to rotate, and rotates the whole object about the x_3 axis to enable planar bending deformation in all direction. Twist $\mathbf{t}_2(\mathbf{x})$ rotates the block back to its original orientation. Bending deformation involves rotation and displacement of the top plane. These two deformations provide the base case of modeling. Stretch deformation can change the arclength of the backbone curve. It is also essential to include shear deformation to add more DOFs. Composition operation combines various deformations and generates a complex deformation with sufficient degrees of freedom while satisfying the boundary condition:

$$\mathbf{f}(\mathbf{x}) = \mathbf{t}_2(\mathbf{b}_2(\mathbf{s}(\mathbf{e}(\mathbf{t}_1(\mathbf{x}))))) \quad (29)$$

The scalar functions defining each deformation type are expanded as a weighted sum of modes in Table II.

$ro, tw, st, s_1, s_2, b_1, b_2, b_3, b_4$ are the mode parameters. Only 9 DOFs are involved in the composite deformation, which keeps the modeling simple while representing sufficient DOFs. The same stretch deformation is also used to model the necking of a thin film. This case shows the ability of our method to handle very large nonlinear deformations.

IV. RESULT AND DISCUSSION

This section presents the performance of our method by comparing with FEM and physical experiments. Fig. 4 shows the three-way comparison of the results in percentage error. It can be observed that ABAQUS and primitives model have similar percentage errors when benchmarked against physical experiments. During the comparison with ABAQUS, we observe that our method has a larger error at the surfaces of the object. This is due to the assumption in our method that horizontal planar sections remain plane while the real objects usually not preserve these planes. Even so, our method still presents similar error compared to error from ABAQUS.

The method is also compared with SOFA, an open-source framework for real-time dynamic FEM simulation. SOFA performs similar to our method for the case with smaller deformation, while our method also provides more accurate result for case with large deformation (the film case), as shown in Fig. 5.

The above results show that our method renders comparable coordinates as FEM nodes. FEM is based on discretizing a continuous problem into a finite number of elements, the discretization error is dependent on mesh size and quality. In contrast, our method is mesh-free and can generate the position of any point on the object since the solution is in closed-form.

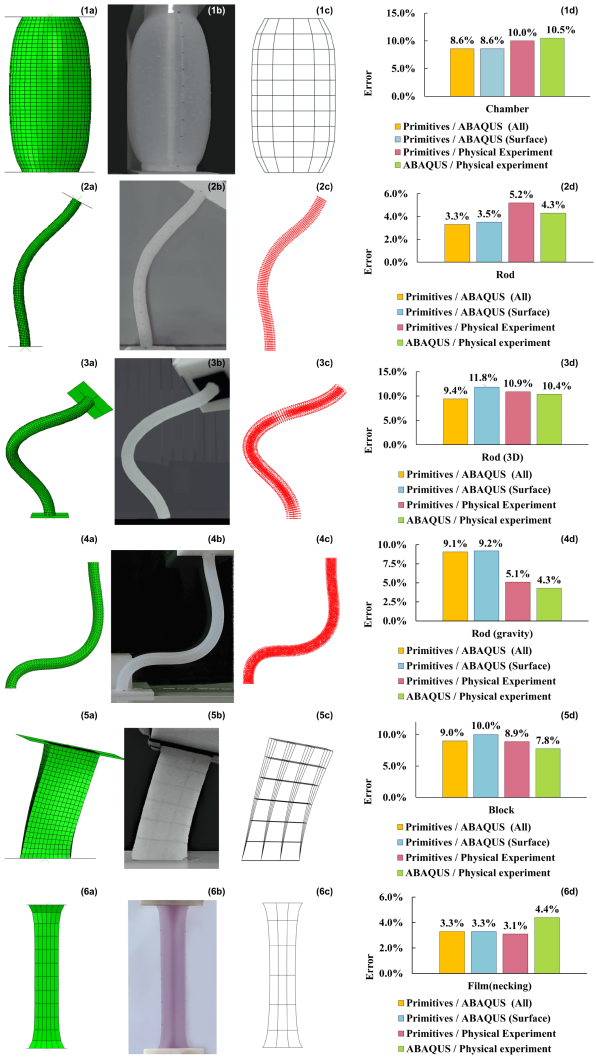

Fig. 4. Experiment result of 1. chamber; 2. rod in 2D; 3. rod in 3D; 4. rod with gravity; 5. block; 6. film. (a) The simulation result from ABAQUS. (b) A photo shot by Zed camera during the experiment. (c) The simulation result from our method. (d) The comparison between two simulation results and experiment. "All" in the legend means the comparison between primitive model and ABAQUS model is about all the nodes. "Surface" means only the points on the object surface are involved in the comparison.

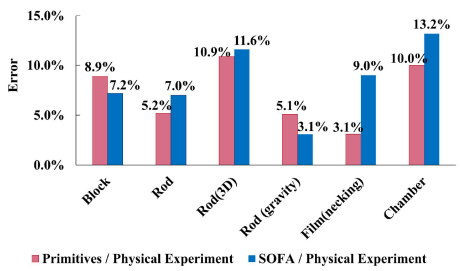

Fig. 5. Error of SOFA framework and our method.

Moreover, our method is significantly faster than ABAQUS, as illustrated in Table III, the primitive method can be 50 times faster than ABAQUS in all cases. Since ABAQUS has not been designed to do real-time computation, the comparisons of computation time with ABAQUS are only indicative of the relative computational complexity of our method vs. FEM, and not the run time when specialized FEM methods are run on multiple GPUs. SOFA is a dynamic simulation tool, so the time measured is the time taken for the object to stabilize at the desired configuration.

TABLE III
RUN TIME OF ABAQUS, SOFA, AND OUR METHOD

Object	ABAQUS time	SOFA time	Primitives time
Chamber	8.2s (3267 nodes)	1.4s (515 nodes)	0.2s
Rod(2D)	35s (6161 nodes)	1.6s (405 nodes)	0.7s
Rod(3D)	32s (6161 nodes)	2.0s (405 nodes)	0.3s
Rod(gravity)	58s (6161 nodes)	2.4s (405 nodes)	1.3s
Block	12s (4225 nodes)	0.7s (637 nodes)	0.1s
Film	2.5s (325 nodes)	0.8s (182 nodes)	0.1s

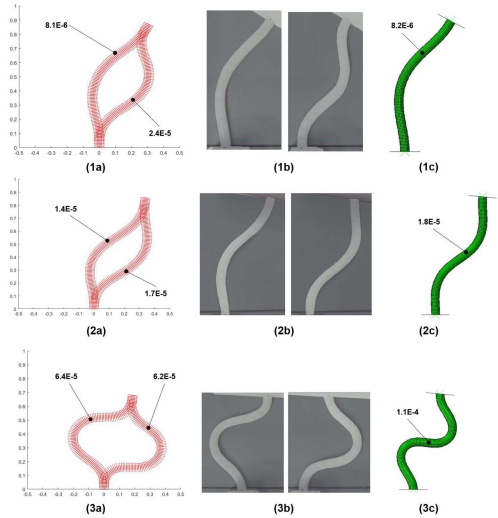

Fig. 6. Configuration generated by ABAQUS and our method. a is the result from primitive model, b is the upper-hand and lower-hand configuration of a real rod in the physical experiment, and c is the result from ABAQUS. The strain energy of these simulation results are annotated in the figure. In series 1, ABAQUS converges to the lower energy branch. In series 2, ABAQUS gives the higher energy branch result. Series 3 shows the case when ABAQUS is trapped in local minimum with significant error. In all series our method generates results from both branches.

Another phenomenon is observed during the rod case. The inverse kinematics problem for a redundant manipulator usually consists of a series of solution branches. Our modeling method could generate results from different solution branches by changing the initial position subtly, while ABAQUS only gives one result, and sometimes trapped in a local minimum. Fig. 6 gives a few examples when our method and ABAQUS give the same or different solution branches.

We also modeled a fiber-reinforced actuator, as shown in Fig. 7. The actuator is primarily composed of rubber material that has been reinforced with helical fibers. One of these fibers is a larger, more prominent helix that is embedded in the rubber at a pitch of 0.8 cm. This larger helix is responsible for inducing twisting motion in the actuator when it is inflated. In addition to the larger helix, there is also a smaller, circular fiber that is positioned around the body of the actuator. This fiber has a very small pitch, which means that it is wound very tightly around the actuator. The purpose of this fiber is to constrain the radial expansion of the actuator so that the actuator can only twist and

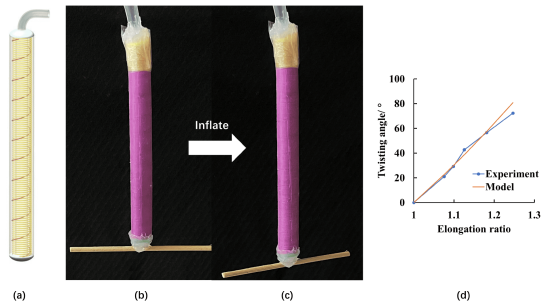

Fig. 7. (a) Structure of a twisting actuator with a network of fibers covering the whole body. Red curves represent the fiber leads to the twisting motion, yellow curves represent the radial constraint that limits the expansion in the radial direction. (b) Actuator before inflation. (c) Actuator after inflation. (d) Experimental results compared with our model.

extend when inflated. The actuator is inflated with a syringe. A horizontal rod is attached to the end of the actuator so the twisting angle is easier to observe. Then a Zed camera is placed below the actuator, and measure the twisting angle and elongation of the actuator. Our method handles high pitch fiber reinforcement well, which cannot be directly compared to FEM since such methods are not built for meshing individual fibers embedded in a 3D continuum.

V. Conclusion

This letter introduces the modeling of soft structure deformation using closed-form locally volume-preserving primitives with boundary conditions applied, as is often the case in soft robotics problems. This method simulates the robot deformation at a global level, and it generates the coordinates in terms of closed-form deformations. This technique is tested on an inflatable chamber, a rod, a film, and a block. As a nonisotropic example, a tube helically reinforced with an inextensible fiber is modeled. Moreover, the applicability of the new method to cases involving gravity loading, which is no longer a purely kinematic constraint, is demonstrated. In all cases, our method has similar speed with SOFA and at least 50 times faster than ABAQUS, while providing comparable results with an error of 10% compared to FEM and physical experiment. This technique could thus be a reduced-order tool to store and reconstruct the output data from FEM, or a time-efficient alternative to FEM in real-time control.

Acknowledgment

The authors appreciate Lyu Shengnan's sharing of valuable ideas regarding modeling of fiber reinforcement.

References

[1] G. S. Chirikjian, "Closed-form primitives for generating locally volume preserving deformations," *J. Mech. Des.*, vol. 117, no. 3, pp. 347–354, 1995.
[2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
[3] H. Lipson, "Challenges and opportunities for design, simulation, and fabrication of soft robots," *Soft Robot.*, vol. 1, no. 1, pp. 21–27, 2014.
[4] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.
[5] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.
[6] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Adv. Robot.*, vol. 26, no. 7, pp. 709–727, 2012.
[7] P. Polygerinos et al., "Towards a soft pneumatic glove for hand rehabilitation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1512–1517.
[8] J. Allard et al., "Sofa-an open source framework for medical simulation," in *Proc. Med. Meets Virtual Reality*, 2007, pp. 13–18.
[9] 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, 2016.
[10] D. Marinkovic and M. Zehn, "Survey of finite element method-based real-time simulations," *Appl. Sci.*, vol. 9, no. 14, 2019, Art. no. 2775.
[11] O. Comas, Z. A. Taylor, J. Allard, S. Ourselin, S. Cotin, and J. Passenger, "Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA," in *Proc. Int. Symp. Biomed. Simul.*, 2008, pp. 28–39.
[12] M. Garcia, C. Mendoza, L. Pastor, and A. Rodriguez, "Optimized linear FEM for modeling deformable objects," *Comput. Animation Virtual Worlds*, vol. 17, no. 3–4, pp. 393–402, 2006.
[13] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2550–2555.
[14] C. Duriez and T. Bieze, "Soft robot modeling, simulation and control in real-time," in *Soft Robotics: Trends, Applications and Challenges*. Berlin, Germany: Springer, 2017, pp. 103–109.
[15] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3982–3987.
[16] S. Niroomandi, I. Alfaro, D. Gonzalez, E. Cueto, and F. Chinesta, "Real-time simulation of surgery by reduced-order modeling and X-FEM techniques," *Int. J. Numer. Methods Biomed. Eng.*, vol. 28, no. 5, pp. 574–588, 2012.
[17] P. Polygerinos et al., "Modeling of soft fiber-reinforced bending actuators," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 778–789, Jun. 2015.
[18] B. A. Jones, R. L. Gray, and K. Turlapati, "Three dimensional statics for continuum robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 2659–2664.
[19] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 490–513, 2020.
[20] D. Kim et al., "Review of machine learning methods in soft robotics," *PLoS One*, vol. 16, no. 2, Feb. 2021, Art. no. e0246102.
[21] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *Proc. IEEE Int. Conf. Soft Robot.*, 2018, pp. 39–45.
[22] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaav1488.
[23] R. L. Truby, C. Della Santina, and D. Rus, "Distributed proprioception of 3D configuration in soft, sensorized robots via deep learning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3299–3306, Apr. 2020.
[24] K. Elgeneidy, N. Lohse, and M. Jackson, "Bending angle prediction and control of soft pneumatic actuators with embedded flex sensors–A data driven approach," *Mechatronics*, vol. 50, 2018, pp. 234–247.
[25] G. S. Chirikjian and A. B. Kyatkin, *Harmonic Analysis for Engineers Applied Scientists: Updated and Expanded Edition*. Mineola, NY, USA: Courier Dover Pub., 2016.
[26] G. S. Chirikjian and J. W. Burdick, "Kinematically optimal hyper-redundant manipulator configurations," *IEEE Trans. Robot. Automat.*, vol. 11, no. 6, pp. 794–806, Dec. 1995.
[27] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 343–354, Jun. 1994.
[28] J. S. Kim and G. S. Chirikjian, "Conformational analysis of stiff chiral polymers with end-constraints," *Mol. Simul.*, vol. 32, no. 14, pp. 1139–1154, 2006.
[29] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.