

Received September 10, 2021, accepted October 1, 2021, date of publication October 8, 2021, date of current version October 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119009

MOOSE-Based Finite Element Hyperelastic Modeling for Soft Robot Simulations

KEVIN WANDKE^{1,2} AND Y Z^{1,2,3}, (Member, IEEE)

¹Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA

²Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA

³Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA

Corresponding author: Y Z (zhyang@illinois.edu)

ABSTRACT While soft robotics is an emerging frontier of the robotics field, accurate modeling of large deformations for soft hyperelastic materials remains a challenge. Herein, we build on the existing open-source Multiphysics Object Oriented Simulation Environment (MOOSE) to accurately model neo-Hookean hyperelastic materials under user-defined loads and large strains. Excellent agreement between the simulated and theoretical results is obtained for simple geometries. Next, we show that our application can accurately model and predict the response of a real soft pneumatic actuator. To conclude, we demonstrate that the open-source nature of our simulation environment enables direct control over simulation parameters, allowing users to tailor the accuracy, convergence, and speed of their simulations.

INDEX TERMS Finite element analysis, open-source software, soft robotics, solid modeling, systems simulation.

I. INTRODUCTION

In recent years, developments in the field of soft robotics have greatly enhanced the scope of work that robots can perform. As a result of their functionally infinite degrees of freedom, inherently compliant nature, and high power-to-weight ratios, soft robots are now seeing use in the areas of manipulation, human-robot interaction, and minimally invasive surgery [1]–[3]. Despite these recent advances, there remain several factors inhibiting the wider adoption and use of soft robots. One such factor is the need for specialized software to perform modeling and simulation on these highly nonlinear multiphysics systems (often requiring the simulation of solid, contact, and fluid mechanics, as well as system dynamics). Traditional robots are readily modeled as rigid bodies using analytic methods, or numerically using efficient and widely available Computer Aided Design (CAD), Finite Element Method (FEM), and simulation softwares [4], [5]. These tools allow traditional robots to be designed and tested in virtual environments, greatly reducing the time, expertise, and cost required for development. In contrast, tools capable of simulating soft robots are often expensive, inaccurate, or have limited feature sets.

The associate editor coordinating the review of this manuscript and approving it for publication was Luigi Biagiotti¹.

Some effort has been made to simulate soft robots using analytic constant-curvature methods [6], [7] or as hyper-redundant kinematic chains [8]. Other analytical models have been developed for specific purposes such as trajectory matching [9]. However, despite being computationally efficient, these methods can only model simple loading conditions and are often inaccurate under large deformations or external loading. Others have used novel data-driven approaches to develop models of soft robots [10]. These methods are powerful tools for analysis of systems, but they are often black-box input output mappings, or are domain specific (e.g. control). As a result, these approaches lack the utility of an accurate general purpose model.

More robust attempts to model soft robots make use of FEM software packages. Work on FEM in soft robotics often tries to predict and optimize the design of soft actuators, as seen in [11]. More specifically, work has already been done using COMSOL-Multiphysics to model the nonlinear nature of soft sensors [12]. ANSYS was used to model hyperelastic material responses and additional soft sensors [13] and explore and optimize a variety of actuator parameters [14]. ANSYS was also used to further characterize the responses of soft actuators, including frictional self-contact in [15]. ABAQUS has also been used to model the complex electro-mechanical behavior of hyperelastic soft sensors [16], to analyze multi-material fiber-reinforced pneumatic

actuators [17] and to assist in the automatic design and modeling of soft actuators [18]. In addition to these works, a comprehensive review of the current state of soft robotic modeling is given in [19]. COMSOL, ANSYS, and ABAQUS are all powerful simulation platforms that are capable of accurately modeling soft robots. However, they are closed-source commercial softwares. Thus, users of these platforms are unable to implement their own features or changes easily and the degree of control end users have over the simulation is limited. In addition, these software solutions come with costly licensing fees which inhibit broader access.

Another common modeling platform is the SOFA framework and its soft robotics plugin, which was created with the express purpose of modeling soft robots. Furthermore, SOFA is open-source and free to use, allowing a broad audience to use and customize the framework to implement useful features. However, while a powerful tool for soft robotic simulation, SOFA's soft robotics plugin has several drawbacks, one of which is that it employs non-linear geometric models under a linear elasticity assumption [20]. Thus, while hyperelastic models have been implemented in SOFA, they are usually used to model robots undergoing large displacements, but small deformations [21]–[23]. This limits the ability of the system to accurately model highly deformed configurations, which are commonplace for soft robotic systems.

In this paper, we present the development, implementation, and testing of an open-source soft robotics application that enables hyperelastic modeling, resolving one of the major issues presented in the SOFA framework. We achieved this by building off of Idaho National Laboratory's (INL) Multiphysics Object Oriented Simulation Environment (MOOSE). First, we will give an overview of the MOOSE framework and the neo-Hookean hyperelastic model used in this paper. Next, a comparison between the theoretical and simulated results is presented to show the accuracy of the model in simple loading conditions. After demonstrating the accuracy of the approach, we show the capability of the framework to predict the motion of a prototypical pneumatic actuator. To conclude, the open-source nature of our application is leveraged to demonstrate how changes to the time stepping scheme, time integration scheme, preconditioner, solver, and model parameters allow the user to enhance the convergence, computational cost, and accuracy of their simulations. The source code for this project is available at: <https://github.com/Z-Laboratory/Kraken>.

II. METHODS

A. MOOSE ARCHITECTURE

To simulate soft robots, a FEM software with two primary characteristics is required. Firstly, the FEM software needs to offer a high degree of customization so that users can readily implement and change models, features, and parameters. Secondly, the software needs to be able to cope with multiphysics simulations that would potentially include solid

mechanics, dynamics, fluid dynamics, contact mechanics, or any combination thereof.

Given these requirements, the MOOSE framework developed and actively maintained by INL was selected as the most suitable option for the base software. MOOSE is a modular FEM system making use of variable coupling for multiphysics simulation, and allows users to create custom kernels, materials, and applications with ease [24].

MOOSE's Tensor Mechanics module already contains some of the machinery required to run FEM simulations of soft robots. As shown in Figure 1A, within this module strains are computed from displacement data, then multiplied by an elasticity tensor. This procedure produces a stress tensor, which can then be used to calculate displacements. The process is iterated until a solution within a user defined margin of error is obtained.

Within the original MOOSE algorithm, stress is calculated using linear elasticity, which linearly relates the stress and strain of a material with a static elasticity tensor. However, under large deformations like those routinely experienced by soft robots, linear stress-strain relationships no longer hold. To address this problem, materials undergoing large elastic deformations follow nonlinear constitutive models to describe the stress-strain relationships. These constitutive models are usually referred to as *strain energy density functions* and are usually written as a function of the first three invariants of the left Cauchy-Green deformation tensor (the square of local displacement due to deformation) [25], [26].

Figure 1B shows our implementation of these strain energy density functions in MOOSE. The gradient of the displacement variables is used to calculate a deformation gradient (a linear mapping between a line element in the unreformed configuration to a line element in the deformed configuration). The deformation gradient is then used to compute the left Cauchy-Green deformation tensor, which is in turn subsequently used to compute the stresses. This process is repeated iteratively until the simulation falls below a user-specified margin of error. This new architecture does not implement the linear elasticity tensor, allowing stresses to be directly computed from displacements and nonlinear stress-strain behaviors to be modeled.

B. NEO-HOOKEAN MODEL

In this work, the simplest and most popular hyperelastic constitutive law, the neo-Hookean model, is implemented. The strain energy is modeled as $W = C_1(I_1 - 3)$ for incompressible materials [27], where W is the strain energy density, C_1 is a material constant proportional the shear modulus μ , and I_1 is the first invariant (or trace) of the left Cauchy-Green deformation tensor [28].

Taking the derivative of the strain energy density function with respect to the deformation gradient, multiplying by the transpose of the deformation gradient F^T and then dividing by J (the determinant of F and a measure of the local change

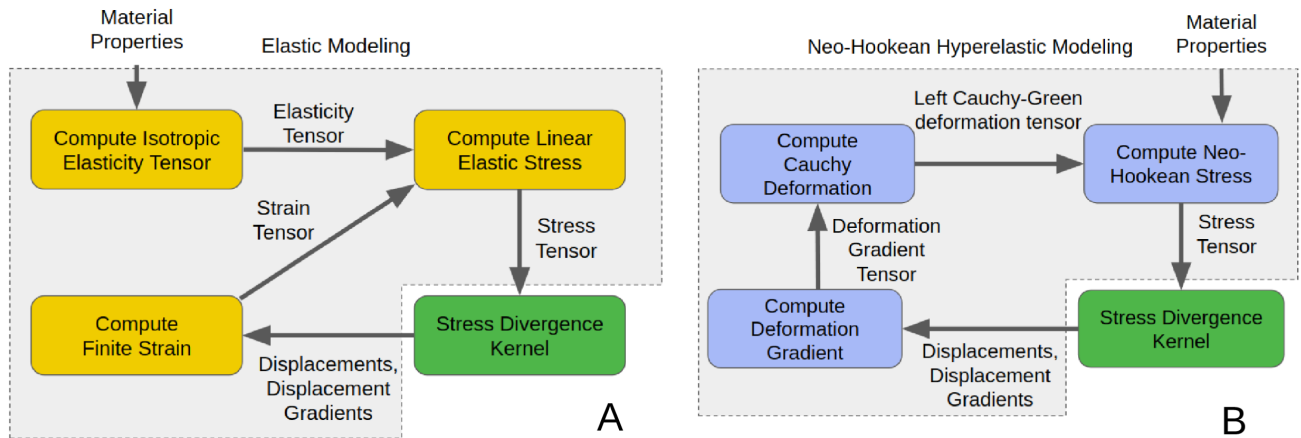


FIGURE 1. Comparison between the built-in elastic modeling architecture in MOOSE and the neo-Hookean hyperelastic modeling architecture implemented in MOOSE. (A) The built-in elastic modeling architecture uses a static elasticity tensor to establish a linear stress-strain relationship. (B) The neo-Hookean hyperelastic modeling architecture computes the Cauchy deformation tensor and then uses a nonlinear neo-Hookean constitutive law to determine stresses.

in volume) yields the stresses on the material:

$$\sigma = \frac{1}{J} \frac{\partial W}{\partial F} F^T \quad (1)$$

For incompressible neo-Hookean materials, this yields the following equation for the stresses [27]:

$$\sigma = 2D_1(J - 1)I + 2C_1B \quad (2)$$

where D_1 functions as a Lagrange multiplier used to enforce incompressibility, I is an identity matrix, and B is the left Cauchy-Green deformation tensor.

Eq. (2) was implemented into a custom material in MOOSE that overrode the default linear stress-strain constitutive equation. MOOSE was then directed to skip the calculation of the strain to improve computation time, as the stress could now be directly computed from the displacements and their respective gradients.

Thus, when a user supplies the constants C_1 and D_1 , the model, built on top of the MOOSE framework will be able to compute the neo-Hookean response of a material with those material properties under large deformations. However, it should be noted that the model is not perfect, and as strains keep increasing our hyperelastic model fails to account for the potential changes in the material parameters under extremely large deformations [29]. Further work will likely include more sophisticated material models such as the Mooney-Rivlin or Ogden models [25], [26], [30], [31].

III. SIMPLE LOADING CONDITIONS

Throughout the validation procedure, we used material constants for Dragonskin 20, which could be obtained from a database of soft material properties [32]. Using the database, we found that the best value of C_1 for our simulations was 0.0822 MPa. This best value was obtained by computing the strain range of our experimental tests and using the database to generate best-fit neo-Hookean parameters for

Dragonskin 20 over that strain range of 0-110%. To improve the model's convergence, we implemented a load stepping procedure where the external pressure boundary condition was set to be $0.5 \times t \times 0.0822$ MPa. To enforce incompressibility, we used a value of D_1 that best matched other silicone rubber's bulk modulus, two orders of magnitude greater than C_1 , or 8.22 MPa. However, given the inexact nature of this approximation, a wide variety of values were tested, without significant impact on the accuracy of results.

A. INCOMPRESSIBLE UNIAXIAL EXTENSION

Once the stress equation, Eq. (2) was implemented in MOOSE, it was important to test how well simulations produced by these equations matched the theoretical results. The first test that was performed was a comparison of the numerical load-strain curves generated by simulations to the theoretical ones. In both simulated and theoretical cases, the incompressible equations were implemented on a (100mm \times 10mm \times 10mm) beam undergoing uniaxial extension.

Prior to obtaining the numerical data, a theoretical stress-strain curve needed to be established. The first step was to rewrite Eq. (2) to give a stress-strain curve for uniaxial extension. Fortunately, these calculations have already been performed [27], [28], and it has been shown that a neo-Hookean material will undergo uniaxial displacement according to the following equation:

$$\sigma = C_1 \left(\lambda^2 - \frac{1}{\lambda} \right) \quad (3)$$

where λ is the stretch ratio, the deformed length divided by the original length. To convert this equation into a stress-strain curve, for every time step in the simulated results, a λ value was calculated. Subsequently, the pressure and the load can be calculated as shown in Figure 2A.

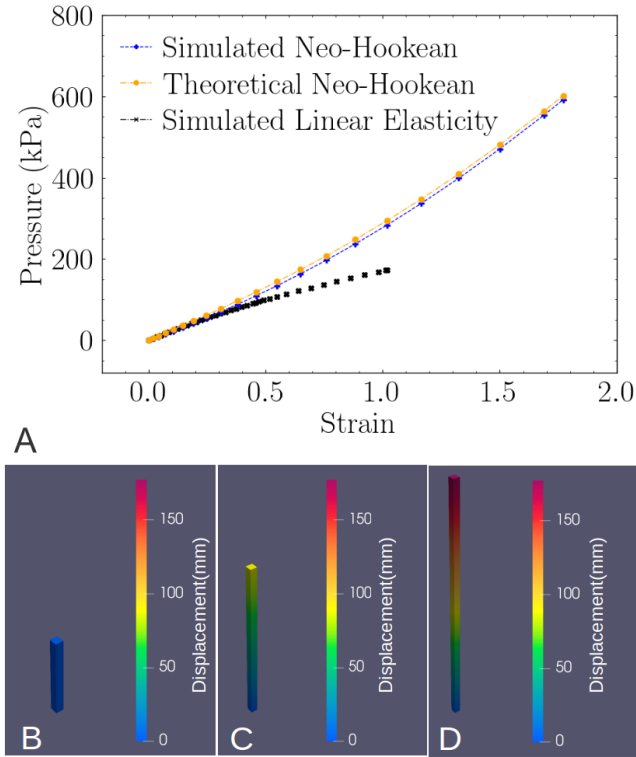


FIGURE 2. Collection of 1-D incompressible uniaxial extension results. (A) Plot demonstrating theoretical and modeled load-strain data. (B) Simulation results prior to loading. (C) Simulation results at 100mm of displacement. (D) Simulation results at the conclusion of the simulation with 192 mm of displacement.

After obtaining theoretical results, the next step was to set up a simulation to evaluate the application’s accuracy. In order to obtain data for a wide range of stresses, a single transient solve was run where the external pressure created by the boundary conditions on the end of the sample increased linearly with time as seen in Figure 2B-D.

Figure 2A demonstrates the agreement between the theoretical and numerical results for the 1-D deformation of incompressible neo-Hookean materials. It also shows the results of our simulation using a linear elastic material as well. It does appear that the hyperelastic simulation slightly underestimates the load that should be required for a sample to reach a desired deformation. The simulation underestimates the stress by a maximum of 10.8 kPa with an average underestimation of 7.8 kPa. Additionally we can see that the simulation of the linearly elastic material fails to converge at a much lower strain, and substantially underestimates the stress at higher strains.

B. INCOMPRESSIBLE BIAxIAL EXTENSION

After verifying that the incompressible neo-Hookean model implemented in MOOSE could accurately simulate uniaxial extensions, it was important to verify that the model was also accurate for more complex loading conditions. An obvious next step was an equibiaxial extension. In this test,

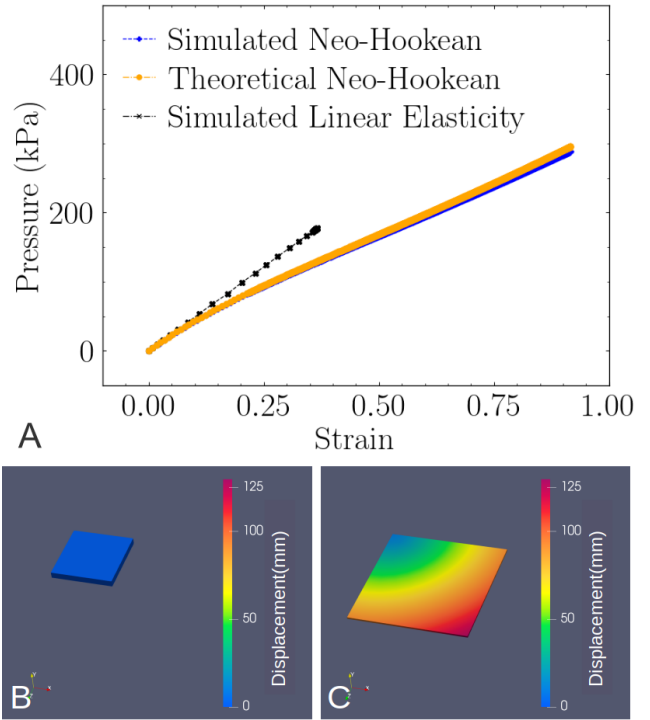


FIGURE 3. Collection of 2-D compressible equibiaxial extension results. (A) Plot demonstrating theoretical and modeled load-strain data, with strain calculated in one of the in-plane directions. (B) Simulation results prior to loading. (C) Simulation results at the conclusion of the simulation with a maximum displacement of 129mm.

a (100mm × 10mm × 100mm) flat plate was modeled to be fixed on two adjacent edges, with a pressure that increased linearly with time applied to the opposite edges. The sample was then pulled equally in both the x and z directions as shown in Figure 3B-C.

This specific loading condition was also selected because theoretical results are straightforward to derive. In this case, the theoretical result is:

$$\sigma = C_1(\lambda^2 - \frac{1}{\lambda^4}) \tag{4}$$

The process described in the uniaxial extension section was repeated with the new theoretical equation, as well as a new mesh geometry and boundary conditions. As shown in Figure 3A, in this test we again see that the simulation follows the theoretical curve quite well, obtaining a maximum underestimation of 7.0 kPa and an average error of 4.1 kPa. This test again indicates that our method is capable of accurately reproducing the theoretical results for incompressible neo-Hookean materials under several loading configurations. However, the linearly elastic material again fails to converge at a lower maximum strain, and substantially overestimates the stress at larger strains.

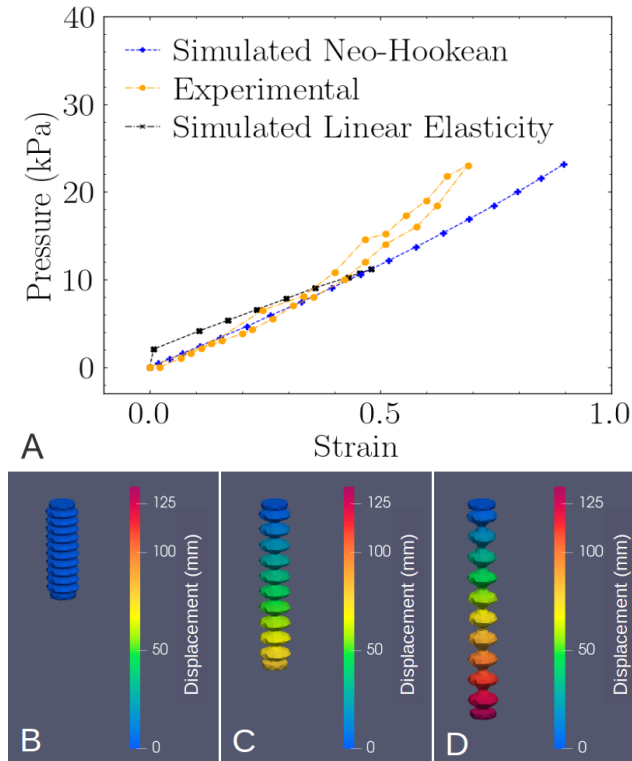


FIGURE 4. Collection of cylindrical actuator extension results. (A) Plot of modeled and experimental pressure-displacement data. (B) Simulation results prior to loading. (C) Simulation results at 65 mm elongation. (D) Simulation results at maximum elongation for simulation convergence, 130 mm.

IV. SIMULATION AND TESTING OF A SOFT PNEUMATIC ACTUATOR

After performing the previous tests and validations, it was established that the numerical method agreed with the theoretical results. Here, we will simulate a prototypical pneumatic actuator and compare against a real-world physical system.

One of the most commonly used components in soft robots is a cylindrical actuator [33]. These pneumatic actuators have four main advantages: radial symmetry, actuation in one direction, homogeneous material properties, and a lack of self-contact. Taken together, these factors suggest that this type of actuator is a good representative system to test and does not require the implementation of a contact mechanics physics engine in the simulation.

As a reminder, both the simulated and experimental tests were performed with Dragonskin 20, with $C_1 = 0.0822$ MPa and $D_1 = 8.22$ MPa. With these parameters, we were able to run a simulation with the internal cavity pressure increasing linearly with time, $p = t \times 0.0822 \times 0.5$ MPa as shown in Figure 4B-D. This parameterization allowed for an internal pressure-displacement graph to be produced as shown in Figure 4A.

After this, the next step was to obtain experimental data from the real system. We designed and manufactured a

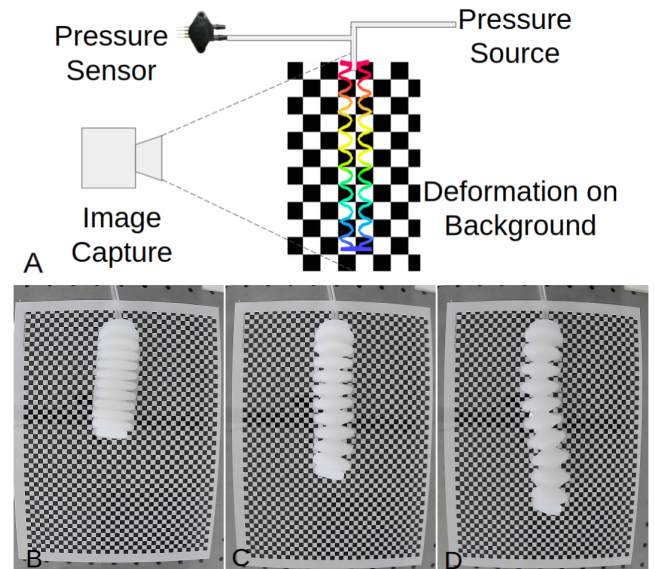


FIGURE 5. Diagram and photos of empirical testing process. (A) diagram of the experimental setup. (B) Soft actuator in relaxed state. (C) Soft actuator in intermediate state. (D) Soft actuator at maximum elongation.

physical system as shown in the diagram displayed Figure 5A and the photos shown in Figure 5B-D. The physical actuator was laid flat on a table and connected to a pressure measurement device as well as a pressure source. By opening a valve connected to the pressure source, the internal pressure of the system increased, and the actuator elongated. An image of the displacement was captured against a textured background, allowing for extension to be calculated as seen in Figure 5C-D. To mitigate friction, the bottom of the actuator was placed on friction-reducing pads.

This setup allowed for a pressure-elongation curve to be obtained for our experimental data. When the experimental data and the simulation data were plotted on the same chart, it was clear that there was agreement between the results at lower strains, but at higher strains the simulation underestimated the pressure (maximum error of 6.2 kPa, average error of 1.4 kPa). One possible explanation for this discrepancy is that the neo-Hookean hyperelastic model does not perfectly describe the material's response at large deformations. Additionally, the simulation using linear elasticity was unable to converge to solutions at far lower strains than the hyperelastic model. Nevertheless, the model we implement in MOOSE is extremely useful to realistically simulate the deformation of soft robots.

V. ANALYSIS OF SIMULATION PARAMETERS

After demonstrating the agreement between the simulated and real world experimental results, it was important to further improve the simulation platform. To do this, the open-source nature of MOOSE was leveraged. While other FEM packages such as ABAQUS, ANSYS, and COMSOL can all model hyperelastic materials, these packages are closed-source and limit the amount of direct

TABLE 1. Comparison of Simulation Parameters.

Test number	Key Parameter	Value	Computation Time (s)	Simulation Time (s)	Strech (%)	Average Error (%)
1	Baseline	N/A	164.6	20.1	394.4	0.18
2	Preconditioner	MOOSE Single Matrix Preconditioner	No Convergence	N/A	N/A	N/A
3	Preconditioner	MOOSE Finite Difference Preconditioner	No Convergence	N/A	N/A	N/A
4	Preconditioner	PETSC LU Decomposition	90.2	30.0	503.4	0.13
5	Preconditioner	PETSC Super-LU Decomposition	98.8	30.0	503.4	0.13
6	Preconditioner	PETSC ASM-LU	90.3	30.0	503.4	0.13
7	Preconditioner	PETSC ASM-ILU	No Convergence	N/A	N/A	N/A
8	Preconditioner	PETSC ASM-BommerAMG	199.7	30.0	503.4	0.13
9	Preconditioner	PETSC adaptivity	259.9	20.1	394.2	0.14
10	Preconditioner	PETSC BoomerAMG	312.7	26.9	471.6	0.13
11	Timestepper	ConstantDT=.1	164.6	20.1	394.4	0.18
12	Timestepper	ConstantDT=.05	180.9	20.1	394.4	0.16
13	Timestepper	ConstantDT=.2	253.5	20.1	394.2	0.13
14	Timestepper	Iteration Adaptive	127.5	20.1	394.4	0.16
15	Timestepper	Soltuion Adaptive	146.8	20.1	394.4	0.16
16	TimeIntegrator	Heun	156.1	30.0	502.9	0.14
17	TimeIntegrator	ARK4	717.4	16.2	343.2	1.93
18	TimeIntegrator	LRK4	279.3	21.0	406.0	0.12
19	TimeIntegrator	Newmark Beta	163.3	20.1	394.4	0.18
20	TimeIntegrator	explicit euler	152.5	30.0	502.9	0.14
21	TimeIntegrator	Central Difference	No Convergence	N/A	N/A	N/A
22	Linear Iteration	10	51.9	2.5	88.8	0.65
23	Linear Iteration	40	349.6	28.8	491.6	0.13
24	Nonlinear Iteration	5	241.2	8.3	221.0	0.22
25	Nonlinear Iteration	20	164.1	21.8	414.5	0.11
26	Threads	1	3045.0	29.8	500.9	0.08
27	Threads	5	374.3	21.8	414.6	0.14
28	Threads	20	179.9	20.1	394.2	0.18
29	Error Tolerance	10x lower	246.3	8.1	218.5	0.21
30	Error Tolerance	10x higher	90.4	20.2	395.3	0.22
31	Compressability	TRUE	235.3	19.6	406.9	-3.98
32	Volumetric penalty	10x higher	No Convergence	N/A	N/A	N/A
33	Volumetric penalty	10x lower	61.8	30.0	503.4	1.57
34	Rotations allowed	TRUE	15.6	1.9	69.8	1.39
35	No scaling	TRUE	44.4	4.0	130.1	0.73
36	Undisplaced Mesh	TRUE	29.4	9.9	134.9	0.33

control users have over the simulation environment. While the simplicity afforded by these packages is usually beneficial to users, in the case of more exotic and extreme simulations such as soft robotics, an open-source package offers several clear advantages. First, an open-source software can be more easily integrated into other simulation, control, or data analysis packages as users have the ability to directly control the types of inputs and outputs generated. Secondly, open-source packages are usually free, making them attractive to a wide range of users who would be unable or unwilling to pay for simulation software. Next, open-source softwares allow users to directly control the simulation parameters, a useful quality for complex simulations such as those often performed in the soft robotics community. A final potential advantage of an open-source framework is that they can often be tailored to specific use cases, and thus can run far more efficiently than more general-purpose softwares.

While all of these factors contribute to the attractiveness of an open-source soft robotic simulation environment, in this section of the paper we intend to more thoroughly investigate the advantage that direct control over simulation parameters can afford end users. We intend to more fully explore both the advantages of integration with other software packages and well as potential gains in efficiency in future works.

The purpose of investigating a variety of simulation parameters is twofold. First, this section seeks to demonstrate that by using an open-source software, users have enormous control over the simulation's behavior, and can alter its performance to meet their needs. Second, this section will provide end users of our software with an understanding of how different simulation parameters can impact the convergence, accuracy, and computational cost of their simulations. Throughout the analysis five types of different

simulation parameters are investigated. These parameters are: the preconditioner used in the Jacobian-Free-Newton-Krylov method, the time-stepping scheme, the time integration scheme, the solving parameters, and the model parameters. These different parameters were selected based off of the significant impact that they had on the quality of the simulation. To assess the performance of changes to the simulation, three factors were considered: the time it took for the simulation to run (in seconds), how much strain the uniaxial rod could experience prior to the model failing to converge (percentage), and the average error between the simulated and theoretical results (percentage).

All tests were performed in a manner similar to the uniaxial extension test in section III on an Intel Xenon E5-2698 V4 2.2Ghz CPU. The maximum internal simulation time was set to 30 seconds, rather than 10, which corresponded to a strain of about 500%. Furthermore, the baseline test was run with the following parameters: Preconditioner=Hypr-BoomerAMG Timestepper=ConstantDT=0.1 Timeintegrator=ImplicitEuler, Linear Iteration=20, Nonlinear Iterations=10, Threads=10, Volumetric Penalty=0.2e8 Error Tolerance=1.0e-4, Scaling=Allowed, Rotations=Small, Mesh Displacement=True. The results can be found in Table 1 above.

From these results we can draw several useful conclusions. Firstly, we can see from Tests 2-10 that the choice of preconditioner has a dramatic impact on both the model's convergence and speed. While the default preconditioner (Hypr-BoomerAMG) is recommended for most users, it can be seen that using a direct preconditioner (the LU family of preconditioners) allows for far better convergence as well as vastly improved speed.

Tests 11-15 show that implementing an adaptive time-stepping scheme can also reduce computational cost, and that a poor choice of initial time-step can increase the cost dramatically. Tests 16-21 show that time integration schemes that are supposed to be more accurate (4th order RK methods) are far more computationally costly and can have reduced accuracy in some cases. Furthermore, explicit time integrators can improve model convergence and speed without sacrificing accuracy.

Tests 22-25 show increasing the number of linear and nonlinear solver iterations for each time step will increase the cost, convergence, and accuracy of the simulation. The exception is increasing the number of nonlinear iterations. This reduces the number of failed time-steps, meaning that increasing this value can lead to decreases in computational cost. Tests 26-28 show that by increasing the number of threads that the process uses, computational time can be decreased. Interestingly, the parallelization is not perfect, and increasing the number of threads comes at a cost to model convergence and accuracy.

Tests 29 and 30 show that by increasing the maximum allowable error, the simulation can run much faster, without sacrificing much accuracy. Meanwhile stricter error tolerance will decrease model convergence. Test 31 shows that it is far

better to use the incompressible version of the neo-Hookean equations as compressible versions performed worse in each performance metric. Tests 32 and 33 show that decreasing the volumetric penalty term can dramatically improve computation time and convergence, but at significant cost to accuracy. Lastly, tests 34-36 show that it is important to use the displaced mesh, automatic scaling, and small rotation parameters for simple tests where large displacements are to be expected.

In summary, the amount of control an open-source simulation platform gives its users allows them to tailor their simulations to closely fit their needs. This feature is not present in many other FEM software packages used for soft robotic simulation, giving this MOOSE-based system a unique advantage over them.

VI. CONCLUSION

In conclusion, this paper outlines a simulation method, implemented in MOOSE, for simulating soft robots using the neo-Hookean hyperelastic model. The ability to accurately predict and model the hyperelastic properties of soft materials within an open-source framework greatly enhances the ability of the soft robotics community to develop robots with specific performance characteristics in mind. Validation was performed by comparing numerical results to the theoretical ones for both uniaxial and biaxial extension loading conditions. Next, MOOSE was used to simulate a soft actuator and the results were compared with the deformation of a real-world soft pneumatic actuator. In all the tests, the model displayed good tracking of the desired result. Finally, a comparison of a wide range of simulation parameters was performed to show how the degree of control over the simulation offered by an open-source framework allows users to customize it to their own needs.

This work of developing an open-source modeling platform capable of predicting the motions of soft robots is vital to the continued development of the field. To date, no such open-source comprehensive platform exists that is capable of implementing hyperelastic multiphysics simulations. Our work will allow researchers to explore a large number of soft robotic designs and controls quickly, and to avoid the costly process of fabricating designs. Currently, without prior knowledge of how designs will interact with their environments, most work is done using a trial and error approach. Overall, this simulation platform will help mature and expand the technology of soft robots to expand the domains in which robots are used.

VII. CODE

The code for this project is available on Github at: <https://github.com/Z-Laboratory/Kraken>

ACKNOWLEDGMENT

The authors would like to thank David Null for his assistance in the design and fabrication of the physical prototypes as well

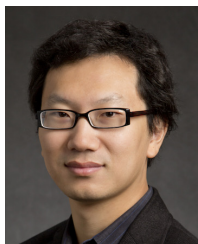
as Alexander Lindsay, Benjamin Spencer, and Daniel Schwen for their assistance on getting started with MOOSE.

REFERENCES

- [1] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: A bioinspired evolution in robotics," *Trends Biotechnol.*, vol. 31, no. 5, pp. 287–294, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167779913000632>
- [2] J. Walker, T. Zidek, C. Harbel, S. Yoon, F. S. Strickland, S. Kumar, and M. Shin, "Soft robotics: A review of recent developments of pneumatic soft actuators," *Actuators*, vol. 9, no. 1, p. 3, Jan. 2020.
- [3] 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, doi: [10.1038/s41598-020-65003-2](https://doi.org/10.1038/s41598-020-65003-2).
- [4] R. Sam, K. Arrifin, and N. Buniyamin, "Simulation of Pick and place robotics system using solidworks softmotion," in *Proc. Int. Conf. Syst. Eng. Technol. (ICSET)*, Sep. 2012, pp. 1–6.
- [5] W. Qian, Z. Xia, J. Xiong, Y. Gan, Y. Guo, S. Weng, H. Deng, Y. Hu, and J. Zhang, "Manipulation task simulation using ROS and Gazebo," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 2594–2598.
- [6] G. Runge, M. Wiese, L. Gunther, and A. Raatz, "A framework for the kinematic modeling of soft material robots combining finite element analysis and piecewise constant curvature kinematics," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 7–14.
- [7] R. K. Katzschmann, C. D. Santana, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. 2nd IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2019, pp. 454–461.
- [8] M. Pozzi, E. Miguel, R. Deimel, M. Malvezzi, B. Bickel, O. Brock, and D. Prattichizzo, "Efficient FEM-based simulation of soft robots modeled as kinematic chains," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4206–4213.
- [9] F. Connolly, C. J. Walsh, and K. Bertoldi, "Automatic design of fiber-reinforced soft actuators for trajectory matching," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 1, pp. 51–56, Jan. 2017. [Online]. Available: <https://www.pnas.org/content/114/1/51>
- [10] D. Bruder, B. Gillespie, C. David Remy, and R. Vasudevan, "Modeling and control of soft robots using the Koopman operator and model predictive control," 2019, *arXiv:1902.02827*. [Online]. Available: <http://arxiv.org/abs/1902.02827>
- [11] Y. Elsayed, A. Vincenzi, C. Lekakou, T. Geng, C. M. Saaj, T. Ranzani, M. Cianchetti, and A. Menciassi, "Finite element analysis and design optimization of a pneumatically actuating silicone module for robotic surgery applications," *Soft Robot.*, vol. 1, no. 4, pp. 255–262, Dec. 2014, doi: [10.1089/soro.2014.0016](https://doi.org/10.1089/soro.2014.0016).
- [12] M. Pozzi, C. Gaudeni, Z. Iqbal, D. Prattichizzo, and M. Malvezzi, "Modeling a sensorized soft layer for adding compliance to the environment in robotic manipulation," in *Advances in Italian Mechanism Science*, V. Niola and A. Gasparetto, Eds. Cham, Switzerland: Springer, 2021, pp. 370–377.
- [13] C. Tawk and G. Alici, "Finite element modeling in the design process of 3D printed pneumatic soft actuators and sensors," *Robotics*, vol. 9, no. 3, p. 52, Jul. 2020. [Online]. Available: <https://www.mdpi.com/2218-6581/9/3/52>
- [14] W. Hu, R. Mutlu, W. Li, and G. Alici, "A structural optimisation method for a soft pneumatic actuator," *Robotics*, vol. 7, no. 2, p. 24, Jun. 2018. [Online]. Available: <https://www.mdpi.com/2218-6581/7/2/24>
- [15] F. Yang, Q. Ruan, Y. Man, Z. Xie, H. Yue, B. Li, and R. Liu, "Design and optimize of a novel segmented soft pneumatic actuator," *IEEE Access*, vol. 8, pp. 122304–122313, 2020.
- [16] J. T. Overvelde, Y. Mengüç, P. Polygerinos, Y. Wang, Z. Wang, C. J. Walsh, R. J. Wood, and K. Bertoldi, "Mechanical and electrical numerical analysis of soft liquid-embedded deformation sensors analysis," *Extreme Mech. Lett.*, vol. 1, pp. 42–46, Dec. 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352431614000666>
- [17] P. Polygerinos, Z. Wang, J. T. B. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of soft fiber-reinforced bending actuators," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 778–789, Jun. 2015.
- [18] 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. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adem.201500503>
- [19] M. S. Xavier, A. J. Fleming, and Y. K. Yong, "Finite element modeling of soft fluidic actuators: Overview and recent developments," *Adv. Intell. Syst.*, vol. 3, no. 2, Feb. 2021, Art. no. 2000187. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202000187>
- [20] *Softrobots Plugin—Soft Robots Plugin for Sofa*. Accessed: Aug. 20, 2021. [Online]. Available: <https://project.inria.fr/softrobot/about-softrobots-plugin/>
- [21] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtécuisse, G. Bousquet, I. Peterlik, and S. Cotin, "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)*, Y. Payan, Ed. Berlin, Germany: Springer, Jun. 2012, pp. 283–321. [Online]. Available: <https://hal.inria.fr/hal-00681539>
- [22] 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. Autom. (ICRA)*, May 2015, pp. 2550–2555.
- [23] 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.
- [24] C. J. Permann, D. R. Gaston, D. Andrš, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, and R. C. Martineau, "MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, vol. 11, Jan. 2020, Art. no. 100430. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2352711019302973>
- [25] M. Mooney, "A theory of large elastic deformation," *J. Appl. Phys.*, vol. 11, no. 9, pp. 582–592, Sep. 1940.
- [26] R. S. Rivlin, "Large elastic deformations of isotropic materials IV. Further developments of the general theory," *Philos. Trans. Roy. Soc. London. A, Math. Phys. Sci.*, vol. 241, no. 835, pp. 379–397, 1948.
- [27] R. Ogden, *Non-Linear Elastic Deformations* (Dover Civil and Mechanical Engineering). New York, NY, USA: Dover, 2013. [Online]. Available: <https://books.google.com/books?id=52XDAgAAQBAJ>
- [28] T. Pence and K. Gou, "On compressible versions of the incompressible neo-Hookean material," *Math. Mech. Solids*, vol. 20, no. 2, pp. 157–182, Aug. 2014.
- [29] A. Gent, *Engineering With Rubber: How to Design Rubber Components*. Munich, Germany: Hanser Pub Incorporated, 1992. [Online]. Available: <https://books.google.com/books?id=dJtnHQAACA>
- [30] R. W. Ogden, "Large deformation isotropic elasticity—on the correlation of theory and experiment for incompressible rubberlike solids," *Proc. Roy. Soc. London. A, Math. Phys. Sci.*, vol. 326, no. 1567, pp. 565–584, 1972.
- [31] M. Hamza and H. Alwan, "Hyperelastic constitutive modeling of rubber and rubber-like materials under finite strain," *Eng. Tech. J.*, vol. 28, no. 13, pp. 2560–2575, 2010.
- [32] L. Marechal, P. Bolland, L. Lindenroth, F. Petrou, C. Kontovounisios, and F. Bello, "Toward a common framework and database of materials for soft robotics," *Soft Robot.*, vol. 8, no. 3, pp. 284–297, Jun. 2021, doi: [10.1089/soro.2019.0115](https://doi.org/10.1089/soro.2019.0115).
- [33] T. Bieze, F. Largilliere, S. H. Chehade, M. S. Lopez, and C. Duriez, *Fetch Mark I Manipulator*. Accessed: Jan. 8, 2021. [Online]. Available: <https://softroboticstoolkit.com/fetch>



KEVIN WANDKE received the B.S. degree in mechanical science and engineering from the University of Illinois at Urbana-Champaign, in 2019, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. He previously worked at the Argonne National Laboratory, as a member of the SULI Program, and at General Electric's Global Research Center as an Intern with the Edison Engineering Program. He was a recipient of the Chancellor's Scholarship and Olsen Award for Excellence in Undergraduate Teaching.



Y Z (Member, IEEE) received the B.S. degree in electrical science and technology from the University of Science and Technology of China, in 2004, and the Ph.D. degree in nuclear science and engineering from the Massachusetts Institute of Technology, in 2010. From 2010 to 2012, he was a Clifford G. Shull Fellow at the Oak Ridge National Laboratory. He is currently an Associate Professor and a Donald Biggar Willett Faculty Scholar with the Department of Nuclear, Plasma, and Radio-

logical Engineering, Department of Electrical and Computer Engineering, Program of Computational Science and Engineering, Center for Biophysics and Quantitative Biology, and Beckman Institute of Advanced Science and Technology, University of Illinois at Urbana-Champaign. He is also the Associate Head of the Department of Nuclear, Plasma, and Radiological Engineering. His research can be summarized in three words: molecules, materials, and machines. On the basic science side, his group synergistically combines and pushes the boundaries of theoretical and computational molecular science and neutron and X-ray scattering experiments. They develop accelerated molecular simulation methods, based on quantum and statistical mechanics and verified by experiments, which can effectively model a wide

range of long timescale phenomena and rare events. The goal is to significantly extend their understanding of the equilibrium and non-equilibrium properties of natural and synthetic materials from the molecular and electronic level and provide accurate predictions and rational guidance to the design, synthesis, fabrication, and utilization of novel materials. Particular emphasis is given to the physics and chemistry of liquids, especially at interfaces, driven away from equilibrium, or under extreme conditions. On the applied research side, leveraging their expertise in novel materials and modeling, his group advances the development of soft robots and human-compatible machines, robots in extreme environments, intelligent control algorithms, and understandable artificial intelligence that can lead to immediate societal impact. He has been recognized with several awards, including most recently, the Dean's Award for Excellence in Research and the American Nuclear Society Landis Award. He is an Associate Editor of *Science and Technology of Advanced Materials*.

...