

Simulation of Hyper-Elasticity by Shape Estimation

Christopher-Denny Matte and Tsz-Ho Kwok*

Department of Mechanical, Industrial and Aerospace Engineering
Concordia University
Montreal, QC H3G 1M8, Canada

The simulation of complex geometries and non-linear deformation has been a challenge for standard simulation methods. There has traditionally been a trade-off between performance and accuracy. With the popularity of additive manufacturing and the new design space it enables, the challenges are even more prevalent. Additionally, multiple additive manufacturing techniques now allow hyperelastic materials as raw material for fabrication and multi-material capabilities. This allows designers more freedom but also introduces new challenges for control and simulation of the printed parts. In this paper, a novel approach to implementing non-linear material capabilities is devised with negligible additional computations for geometry-based methods. Material curves are fitted with a polynomial expression, which can determine the tangent modulus, or stiffness, of a material based on strain energy. The moduli of all elements are compared to determine relative shape factors used to establish an element's blended shape. This process is done dynamically to update a material's stiffness in real-time, for any number of materials, regardless of linear or non-linear material curves.

Nomenclature

χ	System Energy
E	Young's Modulus
$Q\Sigma D^T$	Matrices obtained by singular value decomposition
U	Strain Energy

1 Introduction

The Finite Element Method (FEM) has been used for decades to simulate deformations of solids, flows of liquids and gases, as well as thermal responses of systems. The applications also include the simulation of soft robots, surgical processes, and animation. This has allowed medical fields, exploration, and video games to simulate more realistic behaviors and phenomena. Additionally, this has enabled designers to simulate before needing to fabricate, allowing

for a more freeform understanding of their tasks and solutions. Current research aims to increase the performance of FEM solvers to allow for realistic deformations at real-time speeds. This, in part, is done by simplifying or reducing the computational costs. Current solutions aim to either reduce the computational load by linearizing complex material properties [1, 2] or simplifying the three-dimensional (3D) models [3, 4]. All of these aim to have real-time simulation for their respective applications while maintaining robustness. However, the need for interactive, complex, and non-linear simulation has made current solutions unintuitive or unable to incorporate both accuracy and speed.

New methods of simulation have been proposed to increase simulation speed, but at the cost of accuracy, such as position and projection dynamics [5]. These simulations ensure visual plausibility while decreasing computational cost. Dealing directly with nodal displacements instead of classical Newtonian methods limits the number of calculations required. Mostly used in computer graphics, they have also been adapted for engineering practices as well for the simulation of smart materials and more intuitive actuation of soft robots [6, 7]. However, these simulations mainly deal with linear phenomena. Some material's behavior depends on their deformed state, meaning their properties and relations need to be updated at every time-step to match their current state. Hyperelastic materials are a class of elastomers that exhibit non-linear material properties and rely on their instantaneous rigidity as a function of strain to dictate their stiffness. This directly applies to soft robots as they are mainly made using hyperelastic material such as silicone, rubbers, or 3D printed polymers [8–11]. This is also true for surgical simulations requiring the modeling of flesh [12]. Their deformation is dictated by the actuation mechanism and the material distribution within the robot. It is currently possible to simulate specific deformations of soft robots given proper constraints and inputs. However, generally, simulations are not fast enough to be all encompassing or able to interact in real time. Some methods proposed faster simulation but failed to depict large deformation accurately or are limited in applications [13].

Recently, Kwok et al. [6] proposed using geometry-

*Corresponding author. Email: tszho.kwok@concordia.ca

defined finite elements (GDPE) to express actuations of smart materials in a geometric form. The traditional FEM, the global matrix in GDPE is not updated during the deformation, which is a key for fast computational speed. This concept was then adapted by Fang et al. [7] to show that actuations typically used for soft robots can be treated as changes in geometry. Their work demonstrated the approach could lead to a real-time simulation of soft robots and significant performance increase over FEA while maintaining the current state-of-the-art stability and speed. Although their work proved that geometry could exhibit the behaviours of two materials simultaneously, it is also the maximum number of materials the method can handle and the property of the materials must be linear. Despite the current limitations, we observed that the method is promising in terms of versatility. It can represent different actuations and material properties by only changing the geometric target shapes. This motivates our research to answer the following research question: how to express the non-linear material behavior through the shape functions on a local level, such that the computational speed is not affected? If it is possible, the framework can also work for multiple materials. Since hyperelastic materials' property changes with strains, they virtually are different materials time-by-time during the deformation. To properly simulate hyperelastic materials, every element needs to be treated individually and have its corresponding stiffness. Thus the number of materials is only limited by the size and density of the mesh. Additionally, since the material properties are dependent on current deformation, the initial material ratios cannot be assumed to be constant at the start of the simulation. The shape blending method prescribes a shape factor for each element based on relative properties, as such the shape factor of one element depends not only on their material characteristics but also on the current state of other elements. This needs to be implemented on a time-step or deformation dependent algorithm to ensure proper estimation of properties. This paper proposes a novel method to address the non-linear material properties and true multi-material capability. Utilizing the GDPE framework, it is possible to translate material models into a relation between two target shapes, a rigid shape based on the previous state of the element and volume conservation of the current shape, that better reflect realistic deformation. This is done by comparing each element's rigidity relative to one another and setting a ratio between a soft configuration and hard configuration. Insight on utilizing multiple materials allows for efficient implementation of hyper-elastic material models within the framework. The contributions are:

1. A mathematical model between the shape factor and material properties is defined and calibrated, which can be applied to convert stiffness into a geometric quantity for simulation.
2. By computing the deformation's strain energy for each element, their instantaneous stiffness can be determined to obtain its corresponding shape factor in each step. Therefore, the framework can virtually work for infinite materials.

3. Hyper-elasticity is incorporated in the geometric computation by controlling each element's shape factor, allowing the global stiffness matrix to stay constant throughout and thus maintaining the fast simulation speed.

The rest of the paper is organized as follows. In Section 2, related works will be presented. Section 3 will outline the geometry-based approach. The methodology of hyperelasticity through geometry is discussed in Section 4. Finally, Section 5 will go over the results and validation of the paper, followed by a conclusion and discussion in Section 6.

2 Related Works

As the complexity of designs and the need for more accurate simulations increase, simulation solutions will have to adapt and increase their performance for non-traditional actuations and use cases. When modelling the human body, many parts can be treated as non-linear materials for more accurate simulation, such as the extensors and tendons in the human hand [12]. Some soft robots also utilize the non-linear behaviour to enable unique deformations [14], while others require accurate, fast simulation for topology optimization [15]. With the complexity of near-infinite degrees of freedom allowed by soft robots, fast and accurate simulations must be developed to enable non-heuristic control [16].

The traditional analysis methods for physical systems involved using Newton's second law to accumulate forces and solve for accelerations. This translates into velocities and, finally, positions of nodes. This process is time-consuming and requires a deep understanding of all the inputs and phenomena happening in the simulation. This has led to the pursuit of different discretization methods and simulation frameworks. There have been some voxel-based approaches [3, 17], others try to simplify the calculations by implementing various forms of mass-spring systems [2, 18]. Model reduction techniques which mapped larger, more complex meshes to reduced meshes, increased performance with minimal loss of accuracy [4, 13, 19]. Although these methods improve the traditional approach in terms of speed, many sacrifice accuracy or robustness in terms of material properties in exchange for visual plausibility.

Work has also been done in terms of material research and quantification. Some research has focused on simplifying non-linear material behaviour and developing new models to reduce the computational cost. Some models have been created using principle stretches to generate spline-based material curves [20]. Others have used linear regression of hyperelastic materials to better estimate the Hessian matrix for optimization, which would reduce the number of times the global stiffness matrix would need to be recomputed [1]. Smith and al. [21] propose a new method to get a more stable energy term from the Neo-Hookean formulation, which results in more stable and probable deformations for hyperelastic material. With the advancement in neural networks and deep learning, non-linear material behaviour has also started

being modelled and studied to reduce computation time [22].

A simpler approach to FEM coined in 2007, position dynamics, was proposed to deal with nodal positions for simulation directly [23]. This would remove much of the calculations and enable a new way of thinking about interactions, dynamics and simulations. A subdivision of position dynamics is the interpretation of the nodes or elements as geometry constraints that can be mixed and solved. This allowed to quickly characterize a designer or modeller's understanding of deformation without requiring in-depth knowledge on what forces would cause this [24]. This was further extended upon with the notion of as rigid as possible deformation, geometric elements that purely focused on maintaining their initial shape [25]. A survey on position-based dynamics highlighted the use of goal positions instead of internal constraints to match initial shapes to deformed configurations [5]. Shape functions for target deformations were introduced for further applications of the geometry-based approach [26]. The approach also led to new optimization frameworks such as the local-global approach [27] which was fine-tuned by Bouaziz et al. [28] and presents new constraints based on geometry. This enabled the simulation of smart materials through a shape matching algorithm [6] and coining of the term Geometry-Driven-Finite-Element. A more interactive and intuitive way of modelling soft robot interactions was developed, showing a performance leap for the state of the art simulation through the help of GDFE [7]. Although these methods provide a new way of defining deformations, they cannot simulate non-linear material behaviours without utilizing traditional FEA methods.

3 Background: Geometry-based Computing

To be self-contained and to facilitate the explanation of our developments, the geometry-based approach [7] is summarized in this section. The approach utilizes a mesh \mathbf{M} which is composed of m elements and n vertices. Each element's shape, a hexahedron for this paper, is described by 8 vertices stored in matrix \mathbf{V}_i . The subscript i represents the element to which these vertices belong. Each element has a desired shape or configuration. This is determined by a blending operation based on the material, which will be detailed later. For clarity, this blended shape will be denoted as \mathbf{V}_i^B and referred to as the desired shape of an element. When constraints or forces are applied to mesh \mathbf{M} , they are transferred into each element, which is then deformed, increasing the energy of the system. The energy is calculated as the difference between the desired shape \mathbf{V}_i^B , and the current shape \mathbf{V}_i defined as

$$\chi(\mathbf{M}) = \sum_{i=1}^m \text{diff}(\mathbf{V}_i^B, \mathbf{V}_i). \quad (1)$$

The algorithm tries to minimize the entire system's energy while obeying all constraints using a two-step solver. The first step is a local step where each element \mathbf{V}_i is used to

orient their corresponding desired shape \mathbf{V}_i^B . A global step then aims to solve an over-constrained linear problem.

3.1 Local Step

There exists a relationship between the current shape \mathbf{V}_i and the desired shape \mathbf{V}_i^B . This step aims to solve for a rotational matrix \mathbf{R}_i which can orient the element's original rigid shape \mathbf{V}_i^R with the current shape \mathbf{V}_i , minimizing the difference in shape and energy. The equation characterizes the difference between both shapes:

$$U = \Omega_i \|\mathbf{N}\mathbf{V}_i - \mathbf{R}_i(\mathbf{N}\mathbf{V}_i^R)\|, \quad (2)$$

where Ω_i represents the weight of the element based on its volume. Typically to match two elements a full transformation matrix would be required. Here, matrix \mathbf{N} is used to center an element and remove the translation aspect of the comparison:

$$\mathbf{N}(i, j) = \begin{cases} 7/8 & \text{if } i = j \\ -1/8 & \text{if } i \neq j \end{cases} \quad \forall i, j \in (1, 2, \dots, 8).$$

\mathbf{N} relates each vertex to one another. In this sense, it is possible to compare the \mathbf{N} matrix to a traditional FEM local stiffness matrix. The transformation matrix \mathbf{T} can then be reduced to a simple rotation and scaling matrix. Considering there are 8 vertices in an element, the problem is over-constrained and can be solved by the least squared method to obtain the affine transformation, including both the scaling and rotation elements. Since the blended shape will enforce the scaling, the only characteristics of interest in \mathbf{T} are the rotational properties. These can be found through singular value decomposition (SVD). This results into $SVD(\mathbf{T}) = \mathbf{Q}\mathbf{\Sigma}\mathbf{D}^T$, where \mathbf{Q} and \mathbf{D} are the rotational components and $\mathbf{\Sigma}$ is the scaling matrix. Matrix \mathbf{R}_i from Eq. 2 can then be defined as $\mathbf{R} = \mathbf{Q}\mathbf{D}^T$, keeping only the rotational elements. Multiplying \mathbf{R} with \mathbf{V}_i^R determines the rigid configuration used in the shape blending process to determine the desired shape \mathbf{V}_i^B .

3.2 Global Solve

The summation of all the local steps creates an over-constrained system of linear equations that can be efficiently solved for shown by this equation:

$$\chi = \sum_{i=1}^m \Omega_i \|\mathbf{N}\mathbf{V}_i - \mathbf{R}(\mathbf{N}\mathbf{V}_i^B)\| = \|\mathbf{A}\mathbf{V} - \mathbf{p}\|. \quad (3)$$

\mathbf{V} is a 3D vector of all the current position of all vertices in the mesh, while \mathbf{p} is a 3D vector of the local solutions for each element. Equation 3 can be solved iteratively [28] by firstly constraining \mathbf{V} and solving for \mathbf{p} , and then constraining \mathbf{p} and solving for \mathbf{V} . This leads to a least square problem in the form of $\mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{A}^T\mathbf{p}$. It is to be noted that matrix \mathbf{A} is composed of solely matrix \mathbf{N} , which means it is constant. This allows for the matrix $\mathbf{A}^T\mathbf{A}$ to be precomputed at the

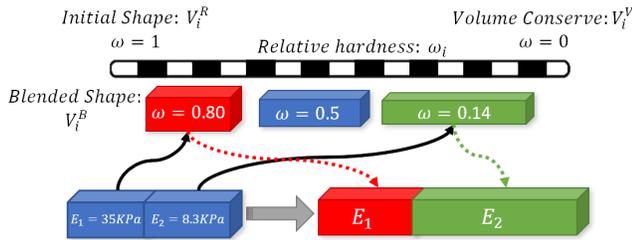


Fig. 1. Visualization of physical properties with relation to shape factor. A larger shape factor results in less deformation, similar to having a larger Young's Modulus.

start of the simulation, making it extremely efficient. Additionally, where typical FEM requires a $3n \times 3n$ for $\mathbf{A}^T \mathbf{A}$, this approach is only $n \times n$ since each axis can be solved individually using the same global matrix. These two properties are part of the reason why this method is attractive for simulation.

3.3 Shape Blending

The geometry-based formulation allows for setting a target shape \mathbf{V}_i^B for each element. In some instances, like as rigid as possible (ARAP) [29], an element will try to stay rigid or similar to its original shape. In others, they might mimic a smart material where they change shape to mimic deformation of a shape memory polymer (SMP) [30]. When there are two different materials in consideration, an element's stiffness is formulated based on how much it cares about its original shape compared to simply conserving its volume. As such, if one element cares about its original shape more than another, it will better maintain its shape during the energy minimization process. In the work of Fang et al. [7], two materials were stretched using a tensile test. They had their relative elongation compared and fitted to a general equation which assigned their relative stiffness using a shape factor ω as depicted in Fig. 1.

Equation 4 depicts how the stiffness of an element i is defined in the geometry-based approach. If the mesh is made of a singular material all ω_i will be equal. If there are two materials, than all elements of the first material will have its corresponding ω_1 , while the second material will have ω_2 , both of which are determined experimentally.

$$\mathbf{V}_i^B = \omega_i * \mathbf{V}_i^R + (1 - \omega_i) \mathbf{V}_i^V. \quad (4)$$

Figure 2 depicts the reasoning behind this approach. Suppose the total strain energy \mathbf{U} given a certain elongation is determined. In that case, it could be reasoned that under uniform loading conditions, two materials with different stiffness would take an equal part in dividing the total strain energy when simulated together $U_T = U_S + U_R$. As such, a rigid material that is twice as stiff as a softer material should have a strain ϵ_R half as large as the other's ϵ_S , while maintaining the area under the curve equal, $U_R = U_S$.

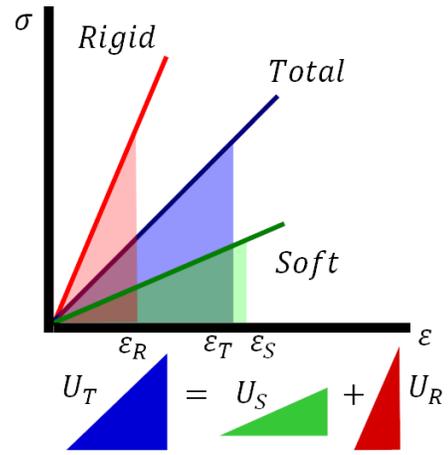


Fig. 2. Multi-material strain energy formulation.

The current limitation is that the calibration method of the previous work only works for two materials. Additionally, the materials have to be linear, as the shape factor is determined and set at the start of the simulation.

4 Methodology

A simple visual approach to calibrating relative material properties has been developed under the assumption the materials behaved linearly [7]. This gave similar results to the real-world examples and demonstrated potential for the shape blending approach. In simple deformation cases under minimal deformation, a linear model for a single material could be used with high fidelity to define deformation. However, hyperelastic materials present non-linear mechanical behaviour and cannot be adequately defined using Hookes law. Due to most materials used in soft robotics, e.g., exhibiting hyperelastic material properties, a new formulation and approach need to be developed to deal with non-linear material models. If the geometry-based approach can extend to make every element have its own material, then it would be possible to model hyperelasticity. However, when dealing with non-linear material models and large deformation, the global stiffness matrix typically needs to be updated to reflect the state of the deformation. This is a computationally expensive task that must be performed after every step of the simulation. Therefore, the challenge here is to accurately reflect material properties and non-linear phenomena without recomputing the global stiffness matrix \mathbf{A} to ensure the performance benefits this method offers persist.

Looking at Eq. 4 it can be seen that the shape factor ω can determine how rigid or soft an element is and directly dictate its stiffness. As such, introducing several materials with different shape factors ω is similar to having elements with different stiffness E , creating a link between geometry and mechanical representations. This allows tuning of an elements properties locally without touching the global stiffness matrix. A rigid element will try to maintain its previous shape during iterations, while a soft element will simply try to keep its volume; thus, its shape may change more freely.

Nevertheless, another challenge is how ω should be defined when there are multiple materials. For hyperelastic materials, the stiffness of a material will change based on its current deformation. As such, the shape factor ω needs to be modified at every step to reflect the non-linear material mechanics properly. Additionally, due to the framework utilizing relative properties, the correlation between elements will have to be updated as well.

If both these concerns are addressed, then any number of materials can be simulated and easily incorporated into the framework. The only metrics that change between materials are the constants used to characterize the non-linear material models; the outputs and inputs are still the same and thus require no additional time for new materials. Two aspects are required to implement non-linear material mechanics to the framework properly:

1. Obtaining the tangent or Young's modulus of a material at any given deformation
2. Efficiently normalize the relative material properties and assign an appropriate ω

4.1 Capturing Hyperelasticity

Material models for hyperelastic materials typically utilize strain-energy density to describe mechanical behaviour and the relation between stress and strain. One of the unique properties of hyperelastic materials is to undergo large strains under relatively low stress and return to their original configuration with minimal plastic deformation [31]. Additionally, a hyperelastic material's properties are dependent on the current deformation of the material [32]. As each element of a discretized model will experience different strain, a unique stiffness is required for each, resulting in different responses to loads and actuations. The phenomenon is further amplified when multiple hyperelastic materials are simulated together as their individual non-linear deformations will be based and compounded relative to one another. This necessitates a more profound understanding of the material's mechanics and how to emulate them better.

The derivation of the strain energy can be obtained by the principal stretches of the deformation gradient [33], which are then used to obtain the tangent modulus of a hyperelastic material required for geometric models. The deformation gradient tensor \mathbf{F} represents all the required information for transforming an element's initial shape to the current configuration. The determinant of \mathbf{F} is the Jacobian \mathbf{J} which represents the change in volume of the element. To obtain the equivalent strain of an element, the deformation gradient can be found by comparing the current shape of the element \mathbf{V}_i and the initial shape \mathbf{V}_i^R .

The process is similar to the local step of the geometry-based framework, but the scaling matrix is of interest here, i.e., $SVD(\mathbf{F}) = \mathbf{Q}\mathbf{\Sigma}\mathbf{D}^T$, with

$$\mathbf{\Sigma} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Taking the principle diagonal of the $\mathbf{\Sigma}$ matrix gives the strain tensors in each axis which are used to calculate the Von Mises equivalent strain:

$$\epsilon_{eq} = \frac{1}{(1+\nu)} \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}{2}} \quad (5)$$

where ν is the Poisson's ratio. The tangent modulus can then be obtained by taking the stress-strain curve's derivative at ϵ_{eq} . Solving for every element's tangent modulus allows for comparing elements and calculating relative material properties, creating the inputs for the second criteria.

4.2 Geometric Modeling of Hyperelasticity

As mentioned in Section 3.3, the stiffness of an element is done within the blended shape by selecting a rigidity shape factor ω_i between a rigid shape \mathbf{V}_i^R and a volume conserving shape \mathbf{V}_i^V . However, this is a geometric representation of stiffness with a constant global stiffness matrix. Material mechanics and physical stiffness are typically handled in a global stiffness matrix that can be updated or modified during a simulation to express non-linear phenomena such as material properties, load distribution or large deformations. The mechanical stiffness is typically reflected through the Young's Modulus, E , of the material. However, the geometric definition of rigidity has a normalized scale from 1 for perfectly rigid, to 0 for an element that solely cares about its volume; requiring mapping to reflect the mechanical implications accurately. By that token, an element with a shape factor ω_i of 0.9 compared to an element with ω_i equal 0.1 will not perform with the naive ratio of being 9 times stiffer.

$$\Gamma(R_M) \mapsto R_G \quad (6)$$

The mapping function Eq. 6 is required for both the mechanical and geometrical stiffness to perform identically. The ratio of the two shape factors (i.e., $R_G = \omega_x/\omega_y$) corresponds to the desired ratio of its physical counterpart (i.e., $R_M = E_x/E_y$), where E_x and E_y are the Young's modulus of two theoretical materials. Once developed, the number of materials the software can handle can be endless. In this sense, hyperelastic elements can also be modelled since every element will have a unique stiffness based on its current deformation.

4.2.1 Mapping

The mapping process involved stretching a rectangular mesh to a pre-defined length. The left half of the mesh has one shape factor ω_{left} , and the right side has another shape factor ω_{right} assigned to it. The input $R_G = \frac{\omega_{left}}{\omega_{right}}$ was set from 1 to 90. For example, $R_G = 30$ means that the left side is 30 times stiffer than the right, which can be achieved by setting the left side to 0.9 and the right side to 0.03. Af-

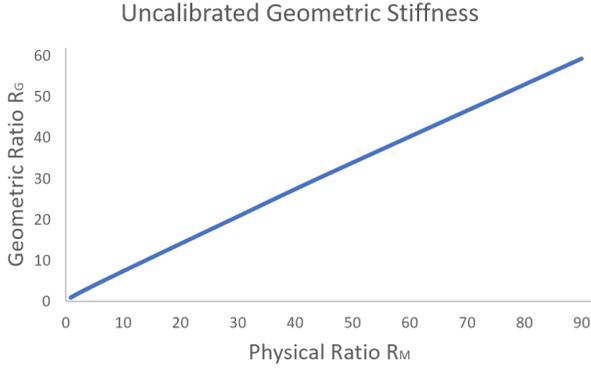


Fig. 3. Calibration of Young's modulus ratio and geometric stiffness ratio.

ter both sides have a shape factor assigned, it is stretched to 100% elongation and the equivalent strain energy of both halves are recorded. The output is the ratio between strain energies and is equivalent to the ratio of physical stiffness R_M . The input for the mapping is R_G , and the strain energy obtained after a simulation is the output R_M . The right side's ω is modified, and the process is restarted. This generates an uncalibrated relationship between the mechanical ratios and geometric ratios for each case, that is plotted and fitted. The results of the uncalibrated simulation are shown in Fig. 3 and demonstrate that the geometric stiffness is stiffer than the desired mechanical response when comparing two linear materials. Comparing the geometric ratio R_G to the desired mechanical ratio R_M showed a non-linear relation between the two. Curve fitting was performed on the data to obtain a correlation. Eq. 7 is used to map the ratio between Young's Modulus R_M , and the equivalent geometric ratio R_G which is used for the simulation.

$$R_G = 0.0017R_M^2 + 1.42245R_M - 0.4141 \quad (7)$$

Elements are given their own relative stiffness ratio for the current step represented by $R_{M,i} = E_{max}/E_i$, where E_{max} is the largest tangent modulus, and E_i is the tangent modulus of an element at its current deformation. Given that the relative properties must be normalized, the maximum and minimum are identified and are at the extremities of the weighting, whose ratio between E_{max} and E_{min} is given by $r_{max} = E_{max}/E_{min}$. To center and normalize the local stiffness of elements around a shape factor $\omega = 0.5$, ratio $h = r_{max}/(r_{max} + 1)$ is derived. Using the above notations and equations, the geometric shape factor ω_i for each element is calculated by

$$\omega_i = \frac{h}{R_{G,i}}, \quad (8)$$

where $R_{G,i}$ is obtained from the calibration Eq. 7. An outline

ALGORITHM 1: Hyper-Elastic Calculation

Input: The rest shape \mathbf{V}^R and current positions \mathbf{V}_i for investigated elements and material.

Number of elements m

Output: The local stiffness ratio ω for blended shape \mathbf{V}^B

```

1 while  $i < m$  do
    /* Calculate Equivalent Strain */
2   Compute  $\mathbf{F}$  between  $\mathbf{V}_i^R$  and  $\mathbf{V}_i$ 
3   Apply SVD to obtain principle strains  $\lambda_{1,2,3}$ 
4   Use Eq. 5 to obtain equivalent strain
5   Record min and max tangent modulus,  $E_{i,min,max}$ 
6 end
7 while  $i < m$  do
    /* Match Stiffness Ratio */
8   Calculate minimum and maximum modulus ratio
     $r$ 
9   Compare element  $E_i$  to  $E_{max}$ 
10  Solve for  $\omega_i$  using Eq. 8
11 end

```

of the process to compute this shape factor is presented in Algorithm 1. It is to be noted that the geometric formulation's least square problem $\mathbf{A}^T \mathbf{A} \mathbf{V} = \mathbf{A}^T \mathbf{p}$ in the global step (3.2) has a constant global matrix $\mathbf{A}^T \mathbf{A}$ regardless of the number of materials. The changes in the shape factor ω_i only update the right-hand side of the equation, i.e., the vector \mathbf{p} . This is because every element already requires its own ω_i , and all the ω_i can be the same or different with no additional computational cost. The only added computation is Algorithm 1 for non-linear materials. This step is equivalent in time to a single local solve and is only performed periodically to ensure an accurate representation of the material properties. Thus, it has a negligible impact on the overall performance.

4.2.2 Integration of non-linearity

Since the current rigidity is based on instantaneous deformation for hyperelastic materials, the converged deformation state of the previous step needs to be reflected in the shape blending algorithm as well. To do this, an intermediary shape \mathbf{V}_i^P can be calculated at each step to represent the previous deformation:

$$\mathbf{V}_i^P = (1 - \alpha) \mathbf{V}_i^R + \alpha * \mathbf{V}_i^v, \quad (9)$$

which denotes the relationship between the current volume conserved shape \mathbf{V}_i^v and the initial rigid shape \mathbf{V}_i^R . The ratio α is defined as $\alpha = \epsilon_p / \epsilon_c$, where ϵ_p is the strain of the previous step, while ϵ_c is the strain of the current shape. This allows the blending method to capture an element's state as a new reference point without overemphasizing the captured shape, thus reducing path dependency and risk of divergence. The more a shape deforms, the lower the impact of the rigid shape, but it also prevents a shape from completely diverging

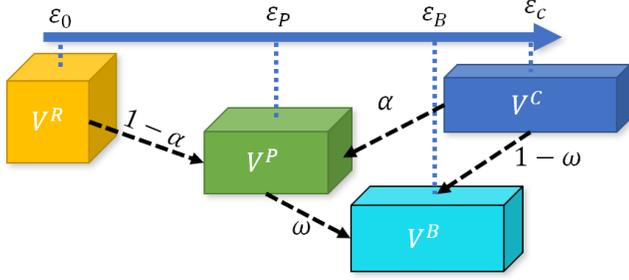


Fig. 4. Visual representation of the blended shape V^B and how it is composed. The previous shape V^P is based on a composition of the rigid shape V^R and Volume preserved shape V^V .

during an iteration as the energy will increase proportionally if it changes too much. A modified version of Eq. 4 incorporates the new metric. The blended shape is then defined as the shape factor ω balancing both \mathbf{V}_i^P and \mathbf{V}_i^V as shown below:

$$\mathbf{V}_i^B = \omega_i * \mathbf{V}_i^P + (1 - \omega_i) \mathbf{V}_i^V. \quad (10)$$

\mathbf{V}_i^B is a blend of the element's initial shape and the deformed shape. Here, α represents the instantaneous state of the element, while ω is the relative stiffness of the element. In this sense, a stiffer material will always try to maintain its shape, while the softer will care more about volume with an approximate shape. An illustration of the relationship between the shapes and physical meaning is shown in Fig. 4.

5 Results

Three tests were performed to validate the framework and its ability to simulate non-linear material properties. The first demonstrates the ability to simulate multiple linear materials simultaneously. The second experiment compared the ability of the software to simulate non-linear materials and its performance. Finally, the last experiment validated the framework's ability to simulate two 3D printed hyperelastic filaments in a tensile test while highlighting linear models' inadequacy for such material. The first and second tests are compared to an industry FEA solver Abaqus. All simulations were performed on the same machine with an Intel i7-7700HQ CPU at 2.80GHz, and 16GB of DDR4 RAM. The framework was coded in C++ and utilizes the open-source Eigen library [34] for solving.

The hyperelastic materials used for validation are Cheetah and NinjaFlex filaments. The physical samples were printed on an Ender 5 printer using the Flexion Extruder. The extruder is specifically designed for direct drive extrusion of flexible filaments. These filaments were then printed into a standard shape to be passed through tensile tests using the ESM 750S. These tests generated the stress-strain curves for each material. The pieces were printed at 100% infill, with 3 walls and grid infill. The printing parameters for the Cheetah were 60mm/s, at 230 degrees, while the Nin-

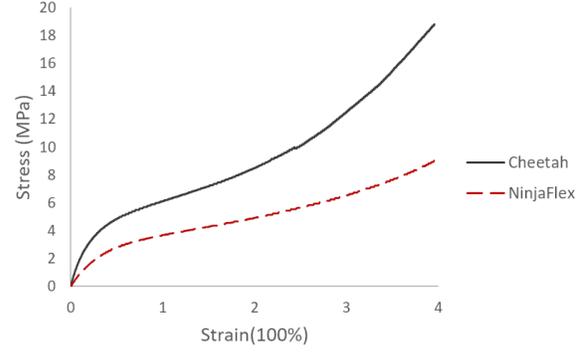


Fig. 5. Stress-strain tensile test data of NinjaFlex and Cheetah filaments

jaFlex was printed at 40mm/s and 240 degrees and 105% flow. Both were printed at 0.2mm layer height. Fig. 5 shows the stress-strain curves for the tensile tests for both NinjaFlex and Cheetah. To capture the material properties from the stress-strain data, it is fitted to different hyperelastic models. The third-order Ogden model [35] was used to fit the data, depicted as follows:

$$W = \sum_{i=1}^3 \frac{2\mu_i}{\alpha_i^2} (\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3), \quad (11)$$

and the constants found for each material is listed in Table 1. This model is used in Abaqus to simulate hyperelastic materials. To allow a direct and simple calculation of the tangent modulus at any given strain to be used in our method, a polynomial was also fit to the test data for each material. The polynomial fitting equation for NinjaFlex and Cheetah are presented in the following respectively.

$$\begin{aligned} \sigma_{Ninja} &= -0.13\epsilon_{eq}^6 + 1.53\epsilon_{eq}^5 - 6.91\epsilon_{eq}^4 \\ &\quad + 15.02\epsilon_{eq}^3 - 15.7\epsilon_{eq}^2 + 8.67\epsilon_{eq} \\ \sigma_{Cheetah} &= -0.11\epsilon_{eq}^6 + 1.49\epsilon_{eq}^5 - 7.69\epsilon_{eq}^4 \\ &\quad + 19.9\epsilon_{eq}^3 - 26.6\epsilon_{eq}^2 + 19.13\epsilon_{eq} \end{aligned}$$

Both the polynomial fit and the Ogden model fit the test data with high similarity. Additionally, both the geometric framework and Abaqus utilized linear Hexadra ("brick") elements.

5.1 Multi-material Linear Comparison

Our work proposes a new method to address multiple materials simultaneously that extends past a two material limit. This is important as hyperelastic materials, when discretized, represent individual non-linear material properties for every element, meaning the number of elements represents the number of individual materials to be simulated. The framework is validated by simulating five different linear materials in bending and comparing it to the same simulation in

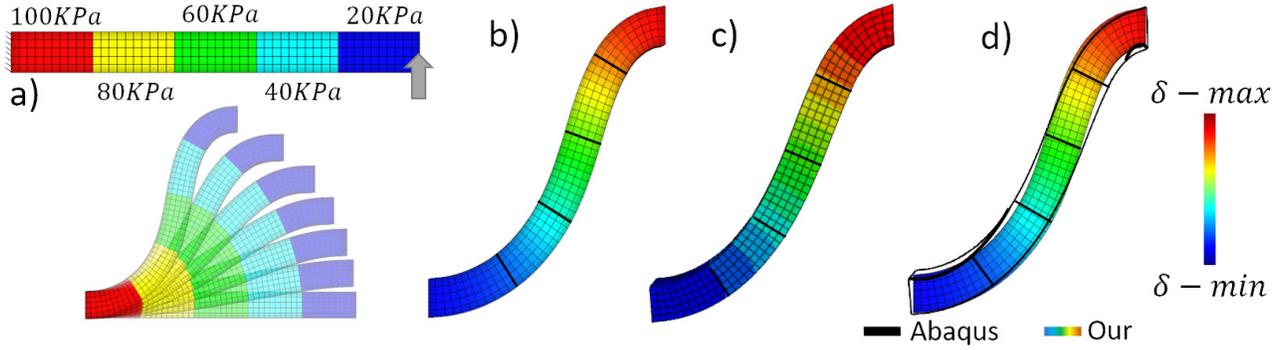


Fig. 6. Multi-material validation for linear materials comparing our framework with Abaqus. a) A $50 \times 5 \times 5$ bar is displaced vertically from the right while maintaining orientation and planerity. The bar is subdivided into 5 different materials with different Young's Modulus; from left to right: 100KPa, 80KPa, 60KPa, 40KPa, and 20KPa. b) Results of our framework; c)Results from Abaqus; d)Overlaid results of our framework in color and Abaqus outlined in black.

Table 1. Ogden Constants for Abaqus

Constants	NinjaFlex	Cheetah
μ_0	0.187297	-5.497378
μ_1	0.091893	2.499418
μ_2	3.397167	12.259373
α_0	1.940480	3.303409
α_1	4.009788	3.722506
α_2	-1.512803	-4.762911

Abaqus, as shown in Fig. 6. A bar measuring $50 \times 5 \times 5$ cm composed of 1250 elements is bounded on the left and displaced vertically on the right. The right-hand side is constrained to prevent rotation but allow horizontal displacement. The bar's material distribution, starting from the left, is: 100KPa, 80KPa, 60KPa, 40KPa, and 20KPa; each representing a unique stiffness. The expected result is for the softer materials to exhibit more significant deformation and curl than the stiffer materials on the left. Fig. 6 d) shows a good correlation with the commercial software. The simulation demonstrates the ability to handle multiple materials without the need to calibrate each one individually. When comparing displacement of every node, a 3% average displacement error is observed. The Abaqus simulation took 71.4 seconds to perform, while our framework took 11.3 seconds. It is to be noted that as the deformation was increasing the convergence time for the Abaqus simulation also increased, demonstrating one of the issues with traditional FEA. Computation time in our framework stayed relatively constant throughout. It has been demonstrated, that the materials' mechanics were directly converted to geometric stiffness and into a shape factor for each material. This example could be repeated with as many simultaneous materials as desired, demonstrating new capabilities compared to previous work.

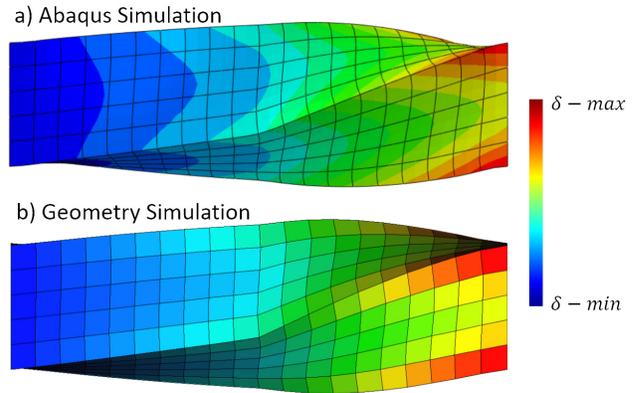


Fig. 7. Rotation comparison between (a) Abaqus and (b) the framework. Left half is composed of Cheetah, right half is composed of NinjaFlex. The colors represent displacement.

5.2 Rotational Comparison

The geometric framework's hyperelastic capabilities were compared to Abaqus in this example. A rotation simulation shown in Fig. 7 was performed where the left half of a bar was characterized as Cheetah, while the right side was formed of NinjaFlex. Both simulations had a size of $20 \times 5 \times 5$ cm and an element count of 500. A 90-degree rotation of the right plane was performed in both Abaqus and the geometry-framework, matching nodes were compared to verify accuracy. An average difference in nodal placement after deformation was noted to be only 3%, with a maximum difference between both simulations of 6%. The Abaqus simulation took 19.35 seconds to perform, while our method took only 4.19 seconds. This demonstrates agreeable results between both cases, while having a significant speed increase.

5.3 Multi-Material Hyperelastic Tensile Test

To validate the material models and the simulation framework for hyperelasticity, a tensile test with multiple materials: Cheetah and NinjaFlex, was performed. Both

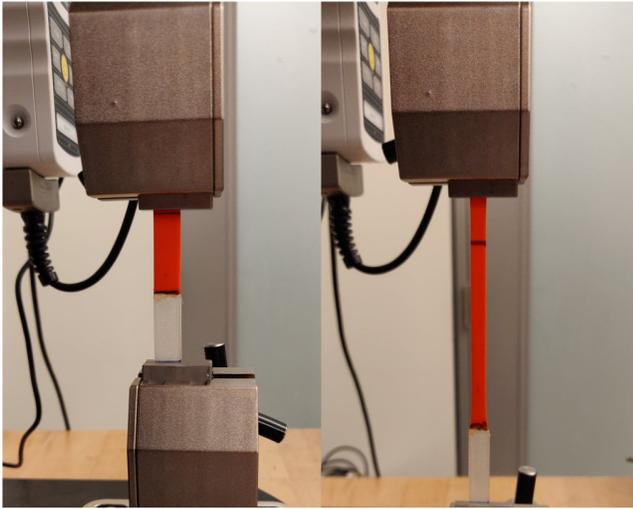


Fig. 8. Experimental set-up for multi-material tensile test. The gauge length of each material is 20mm; NinjaFlex in orange, Cheetah in white.

specimens are joined together thermally and mechanically during the printing process and are of equivalent length. The percent elongation of both materials is measured by keeping track of the joining line between the materials using visual processing. Figure 8 shows the experimental set-up and results. Two simulations were performed to show the effects of hyperelastic properties on a mesh of 500 elements. The first with linear behaviour using the tangent modulus at rest. The second test used the polynomial material curves outlined earlier. The simulation results of both linear and hyperelastic materials are depicted in Fig. 10. A clearer picture can be seen of non-linear material properties when observing Fig. 9, where the strain of the hyperelastic Cheetah seems to be on an up-turn near the end, and the trend of the hyperelastic NinjaFlex is on a down-turn. This signifies the NinjaFlex is at a point of stiffening in its material curve, while the Cheetah's stiffness is relaxing. Compared to the linear simulation, where there is no change throughout the deformation. This kind of interaction becomes all the more prevalent as the number of materials increase. This leads to potentially drastic different configurations for soft robots compared to linear material modelling. This will change the final positioning but also the path the robot will follow.

From Table 2, the difference between the hyperelastic simulation and the experiment is only -1.3% in terms of the location of the material joint, the linear simulation shows a difference of 25.4%. When looking at individual strain, the difference for Cheetah is -10.8%, while the linear simulation shows an error of 112.3%. This can be explained by the quick softening of the NinjaFlex relative to the Cheetah, illustrating the non-linear behaviour. When comparing to the experimental results, Abaqus performed worse than our framework. Through simulating this experiment on Abaqus, it was clear that the interface between materials became problematic when large deformations occurred. Abaqus took 27.2 seconds for the simulation, while our framework took

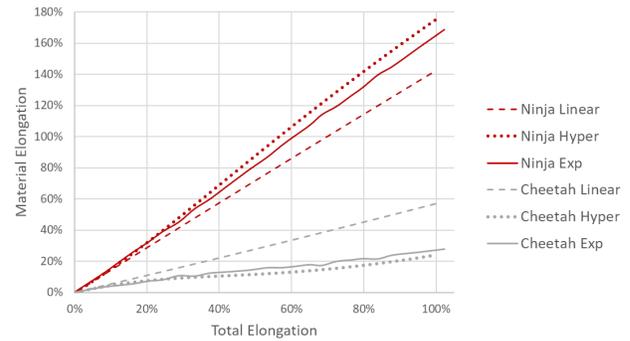


Fig. 9. Graphical representation of the multi-material hyperelastic tensile test. Red depicts NinjaFlex, gray is cheetah.

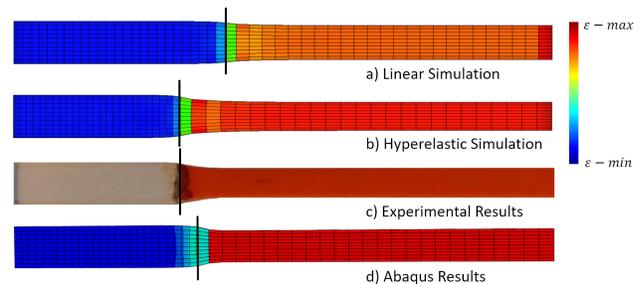


Fig. 10. Cheetah(white) and NinjaFlex(orange) filaments in tension at 100% elongation. a) Linear simulation of our framework; b) Hyperelastic simulation in our framework; c) Experimental results; d) Hyperelastic Abaqus results.

Table 2. Comparison of Multi-Material Tensile Test

Tests	Position (Error)	Cheetah Strain (Error)	NinjaFlex Strain (Error)
Experimental	31.5%	26.8%	168.8%
Linear Sim	39.5% (25.4%)	56.9% (112.3%)	142.3.8% (-15.7%)
Hyperelastic Sim	31.1% (-1.3%)	23.9% (-10.8%)	174.8% (3.6%)
Abaqus Hyper	34.7% (11.6%)	38.8% (62.3%)	161.2% (-7.8%)

18.3 seconds. A 1.49x speed increase is observed. However, the error for Abaqus is 11.6% compared to -1.3% for our method.

6 Conclusion and Discussion

This paper has formulated a novel method for characterizing non-linear and hyperelastic material properties to a geometry-based approach. In this method, material properties can directly and efficiently be integrated into a local shape blending technique dynamically with negligible performance decrease. The calibration for multiple materials

was done based on the equivalent strain energy experienced during uni-axial tensile tests. As such, a direct relation with the mechanical properties of the material could be established and used in the geometry framework. This allowed for a more complete understanding and selection of the shape factor, enabling the direct implementation of material curves. Through the multi-material tensile test, 3D printed specimens were deformed and fitted to obtain material properties. These were compared to a linear model and demonstrated significant differences in results between the two. This further highlights the need for non-linear models for 3D printed filaments and their incorporation in simulation. Compared with standard FEA, the rotational test also showed agreeable results while still maintaining previous performance benchmarks established by the geometry-based approach.

There are, however, some limitations to the current implementation. The current approach uses a polynomial definition of the stress-strain data to obtain Young's modulus from tensile test data. For future consideration, additional material identification tests will be used to obtain a more accurate material model and characterize more complex material behaviours in shear. This would lead to the need for multiple local stiffness metrics that would be balanced to reflect more complex loading scenarios. Additionally, anisotropic properties would require a geometric stiffness for each axis, requiring further consideration in defining a single target shape for an element. An optimization that would discount elements which are not relevant to the actuation and deformation of the model needs to be investigated to ensure the method functions as intended even with non-uniform loading.

Some future considerations would also be investigating using the performance enabled by GDFE to enable design optimization and control of soft robots. Additionally, the possibility of expressing 3D printed parts and their complex geometries and infills as meta-materials to be characterized and implemented in GDFE could lead to the fast computation of optimal material distribution based on actuation and desired deformation. Furthermore, there is research that has developed a formulation for digital materials [36] that are applied on elements directly to assign properties to achieve specific deformations. This can be further extended to an iterative algorithm to achieve material or topology optimization. The current framework would be ideal for dynamically changing material properties with minimal computational overhead.

Acknowledgements

This paper acknowledges the support of the Natural Sciences & Engineering Research Council of Canada (NSERC) grant #RGPIN-2017-06707.

References

- [1] Liu, T., Bouaziz, S., and Kavan, L., 2017. "Quasi-newton methods for real-time simulation of hyperelastic materials". *ACM Trans. Graph.*, **36**(4).
- [2] Duriez, C., Coevoet, E., Largilliere, F., Morales-Bieze, T., Zhang, Z., Sanz-Lopez, M., Carrez, B., Marchal, D., Goury, O., and Dequidt, J., 2016. "Framework for on-line simulation of soft robots with optimization-based inverse model". in *IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, 12, pp. 111–118.
- [3] Hiller, J. D., and Lipson, H., 2012. "Dynamic simulation of soft heterogeneous objects". *CoRR*, *abs/1212.2845*.
- [4] Chenevier, J., Gonzalez, D., Aguado, J. V., Chinesta, F., and Cueto, E., 2018. "Reduced-order modeling of soft robots". *PLOS ONE*, **13**(2), 02, pp. 1–15.
- [5] Bender, J., Müller, M., Otaduy, M. A., Teschner, M., and Macklin, M., 2014. "A survey on position-based simulation methods in computer graphics". *Comput. Graph. Forum*, **33**(6), Sept., pp. 228–251.
- [6] Kwok, T.-H., and Chen, Y., 2017. "Geometry-driven finite element for four-dimensional printing". *Journal of Manufacturing Science and Engineering*, 07.
- [7] Fang, G., Matte, C., Scharff, R. B. N., Kwok, T., and Wang, C. C. L., 2020. "Kinematics of soft robots by geometric computing". *IEEE Transactions on Robotics*, **36**(4), pp. 1272–1286.
- [8] Bartlett, N. W., Tolley, M. T., Overvelde, J. T. B., Weaver, J. C., Mosadegh, B., Bertoldi, K., Whitesides, G. M., and Wood, R. J., 2015. "A 3d-printed, functionally graded soft robot powered by combustion". *Science*, **349**(6244), pp. 161–165.
- [9] U., O. L., and M., M. R. R. D. A., 2013. *Experiments in Underactuated In-Hand Manipulation*, Vol. 88. Springer International Publishing, Heidelberg, pp. 27–40.
- [10] Tavakoli, M., and de Almeida, A. T., 2014. "Adaptive under-actuated anthropomorphic hand: Isr-softhand". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1629–1634.
- [11] Schmitt, F., Piccin, O., Barb, L., and Bayle, B., 2018. "Soft robots manufacturing: A review". *Frontiers in Robotics and AI*, **5**, p. 84.
- [12] Elyasi, N., Taheri, K. K., Narooei, K., and Taheri, A. K., 2017. "A study of hyperelastic models for predicting the mechanical behavior of extensor apparatus". *Biomechanics and Modeling in Mechanobiology*, **16**(3), Jun, pp. 1077–1093.
- [13] Faure, F., Duriez, C., Delingette, H., Allard, J., Gilles, B., Marchesseau, S., Talbot, H., Courtecuisse, H., Bousquet, G., Peterlik, I., and Cotin, S., 2012. *SOFA: A Multi-Model Framework for Interactive Physical Simulation*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 283–321.
- [14] Chen, L., Yang, C., Wang, H., Branson, D. T., Dai, J. S., and Kang, R., 2018. "Design and modeling of a soft robotic surface with hyperelastic material". *Mechanism and Machine Theory*, **130**, pp. 109 – 122.
- [15] Zhang, H., Kumar, A. S., Fuh, J. Y. H., and Wang, M. Y., 2018. "Topology optimized design, fabrication and evaluation of a multimaterial soft gripper". In 2018

- IEEE International Conference on Soft Robotics (RoboSoft), pp. 424–430.
- [16] Largilliere, F., Verona, V., Coevoet, E., Sanz-Lopez, M., Dequidt, J., and Duriez, C., 2015. “Real-time control of soft-robots using asynchronous finite element modeling”. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2550–2555.
- [17] Cheney, N., MacCurdy, R., Clune, J., and Lipson, H., 2013. “Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding”. In GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference.
- [18] Goulette, F., and Chen, Z.-W., 2015. “Fast computation of soft tissue deformations in real-time simulation with hyper-elastic mass links”. *Computer Methods in Applied Mechanics and Engineering*, **295**, pp. 18 – 38.
- [19] Brandt, C., Eisemann, E., and Hildebrandt, K., 2018. “Hyper-reduced projective dynamics”. *ACM Trans. Graph.*, **37**(4), July, pp. 80:1–80:13.
- [20] Xu, H., Sin, F., Zhu, Y., and Barbič, J., 2015. “Non-linear material design using principal stretches”. *ACM Trans. on Graphics (SIGGRAPH 2015)*, **34**(4).
- [21] Smith, B., Goes, F. D., and Kim, T., 2018. “Stable neo-hookean flesh simulation”. *ACM Trans. Graph.*, **37**(2), pp. 12:1–12:15.
- [22] Mendizabal, A., Mrquez-Neila, P., and Cotin, S., 2020. “Simulation of hyperelastic materials in real-time using deep learning”. *Medical Image Analysis*, **59**, p. 101569.
- [23] Miller, M., Heidelberger, B., Hennix, M., and Ratcliff, J., 2007. “Position based dynamics”. *J. Vis. Commun. Image Represent*, **18**(2), pp. 109 – 118.
- [24] Shapira, M., and Rappoport, A., 1995. “Shape blending using the star-skeleton representation”. *IEEE Computer Graphics and Applications*, **15**(2), March, pp. 44–50.
- [25] Sorkine, O., and Alexa, M., 2007. “As-rigid-as-possible surface modeling”. In Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing, pp. 109–116.
- [26] Jin, S., Zhang, Y., and C.L. Wang, C., 2014. “Deformation with enforced metrics on length, area and volume”. *Computer Graphics Forum*, **33**(2), pp. 429–438.
- [27] Liu, L., Zhang, L., Xu, Y., Gotsman, C., and Gortler, S. J., 2008. “A local/global approach to mesh parameterization”. In Proceedings of the Symposium on Geometry Processing, SGP '08, Eurographics Association, pp. 1495–1504.
- [28] Bouaziz, S., Deuss, M., Schwartzburg, Y., Weise, T., and Pauly, M., 2012. “Shape-up: Shaping discrete geometry with projections”. *Comput. Graph. Forum*, **31**(5), pp. 1657–1667.
- [29] Sorkine, O., and Alexa, M., 2007. “As-rigid-as-possible surface modeling”. In Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07, Eurographics Association, pp. 109–116.
- [30] Kwok, T. H., Wan, W., Pan, J., Wang, C. C. L., Yuan, J., Harada, K., and Chen, Y., 2016. “Rope caging and grasping”. In IEEE International Conference on Robotics and Automation (ICRA), pp. 1980–1986.
- [31] Khajehsaeid, H., Arghavani, J., and Naghdabadi, R., 2013. “A hyperelastic constitutive model for rubber-like materials”. *European Journal of Mechanics - A/Solids*, **38**, pp. 144 – 151.
- [32] Renaud, C., Cros, J.-M., Feng, Z.-Q., and Yang, B., 2009. “The yeoh model applied to the modeling of large deformation contact/impact problems”. *International Journal of Impact Engineering*, **36**(5), pp. 659 – 666.
- [33] Chao, I., Pinkall, U., Sanan, P., and Schröder, P., 2010. “A simple geometric model for elastic deformations”. *ACM Trans. Graph.*, **29**(4), July.
- [34] Guennebaud, G., Jacob, B., et al., 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- [35] Ogden, R. W., 1972. “Large deformation isotropic elasticity - on the correlation of theory and experiment for incompressible rubberlike solids”. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **326**(1567), pp. 565–584.
- [36] Leung, Y.-S., Kwok, T.-H., Mao, H., and Chen, Y., 2019. “Digital material design using tensor-based error diffusion for additive manufacturing”. *Computer-Aided Design*, **114**, pp. 224 – 235.