Engineering with Computers
https://doi.org/10.1007/s00366-021-01597-z

ORIGINAL ARTICLE

Inverse deformation analysis: an experimental and numerical assessment using the FEniCS Project

Arnaud Mazier[1] · Alexandre Bilger[1] · Antonio E. Forte[3,4,5] · Igor Peterlik[6] · Jack S. Hale[1] · Stéphane P. A. Bordas[1,2]

Received: 7 June 2021 / Accepted: 23 December 2021

Abstract
In this paper we develop a framework for solving inverse deformation problems using the FEniCS Project finite-element software. We validate our approach with experimental imaging data acquired from a soft silicone beam under gravity. In contrast with inverse iterative algorithms that require multiple solutions of a standard elasticity problem, the proposed method can compute the undeformed configuration by solving only one modified elasticity problem. This modified problem has a complexity comparable to the standard one. The framework is implemented within an open-source pipeline enabling the direct and inverse deformation simulation directly from imaging data. We use the high-level unified form language (UFL) of the FEniCS Project to express the finite-element model in variational form and to automatically derive the consistent Jacobian. Consequently, the design of the pipeline is flexible: for example, it allows the modification of the constitutive models by changing a single line of code. We include a complete working example showing the inverse deformation of a beam deformed by gravity as supplementary material.

Keywords Inverse deformation · Rest position · Undeformed configuration · SOFA · FEniCS Project

1 Introduction

1.1 Motivation

The organization of a standard biomechanical *deformation analysis* pipeline typically proceeds as follows. First, using imaging techniques such as Magnetic Resonance Imaging (MRI) a segmented image of the region of interest is obtained. This segmented image is then meshed so that it can be used as input for a finite-element simulation. The mesh is considered as the initial, undeformed or reference configuration of an elastic body. Then, by applying external forces to this elastic body we can find its deformed (or current) equilibrium configuration.

Conversely, an *inverse deformation analysis* allows us to find the undeformed configuration of a body knowing its deformed configuration. In the case of an object subject to gravity, the undeformed configuration can be seen as a theoretical *gravity-free* configuration. Consequently, determining the rest-position of an organ is of interest in many (bio)mechanical problems, as explained in Fig. 1. For example, in abdominal aortic aneurysms to compute the residual stresses [1, 2]or open-configuration [3] or in breast cancer as an intermedial configuration between the imaging

✉ Stéphane P. A. Bordas
stephane.bordas@me.com

Antonio E. Forte
aeforte@seas.harvard.edu

Jack S. Hale
jack.hale@uni.lu

[1] Institute of Computational Engineering and Sciences, Department of Engineering, Université du Luxembourg, 6, Avenue de la Fonte, 4364 Esch-sur-Alzette, Luxembourg

[2] Institute of Research and Development, Duy Tan University, K7/25 Quang Trung, Danang, Vietnam

[3] Harvard University, 29 Oxford St, 02138 Cambridge, MA, USA

[4] Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy

[5] Faculty of Natural, Mathematical and Engineering Sciences, King's College London, RM 1.26, Strand Campus, London WC2R 2LS, UK

[6] Institute of Computer Science, Masaryk University, Brno, Czech Republic

Published online: 18 February 2022

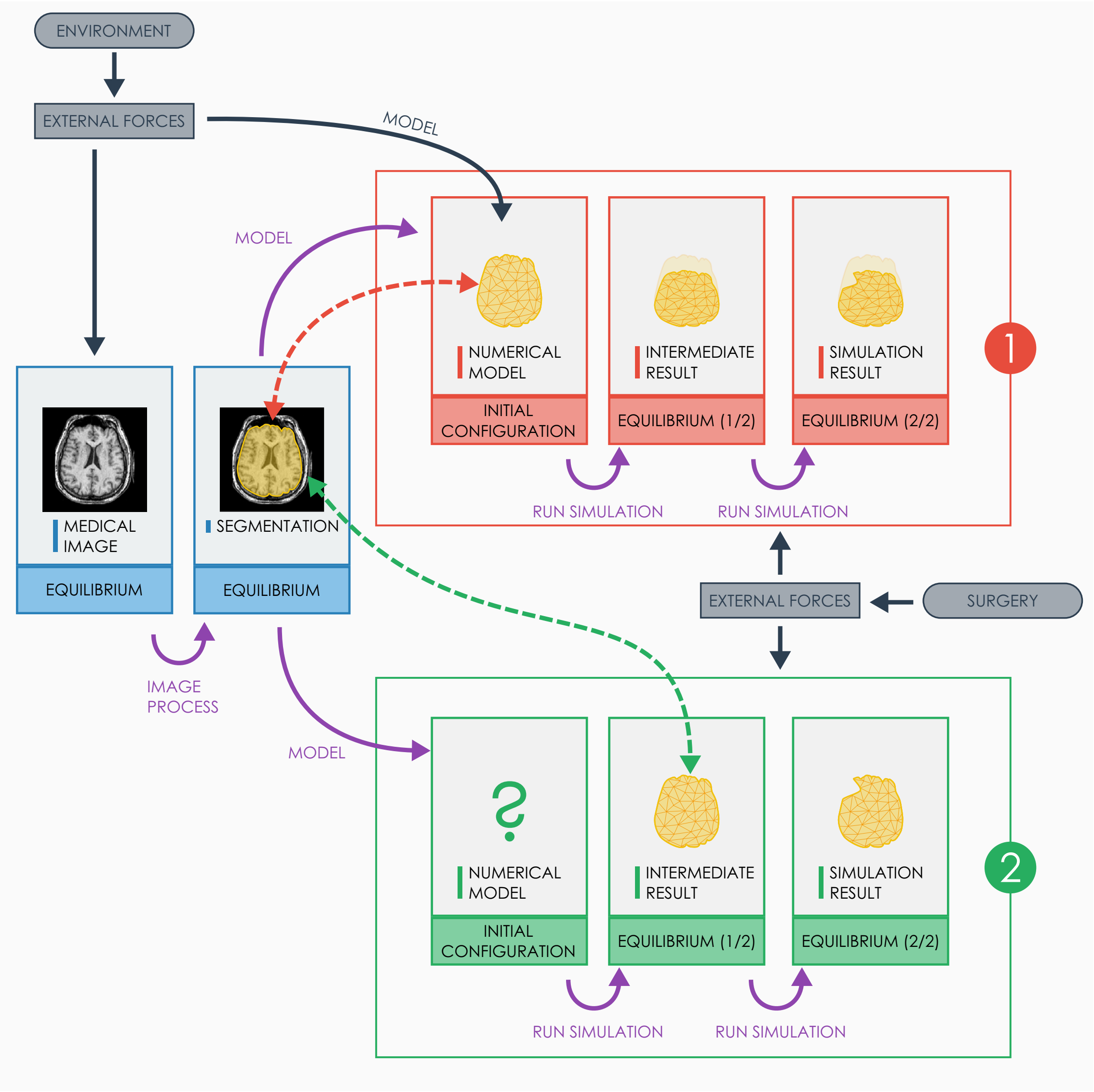


Fig. 1 Our pipeline starts with the acquisition of a medical image, in which the organ can be segmented. The organ is observed at equilibrium under the effect of external forces due to its environment (e.g., gravity). In the usual pipeline (1, in red), the segmented geometry is considered as the initial geometry. Then, the external forces are applied until equilibrium to obtain the intermediate geometry used for simulating the procedure. We propose an alternative approach (2, in green), where we constrain the intermediate geometry to be identical to the segmented geometry. It involves the computation of a new geometry (represented by the question mark symbol), which is the organ geometry such that it would deform to the segmented/imaged configuration if external forces were applied. Here, the final result takes into account the undeformed geometry of the organ

and surgical stance [4]. Besides, this approach can also be used in problems of industrial interest, such as tire or turbine blade design [5, 6].

1.2 Problem statement

The objective of inverse deformation analysis is to determine the undeformed configuration of an object such that it attains a known deformed configuration under the action of a known loading. It is important to note the distinction between inverse deformation analysis and common inverse problems. In a typical inverse problem, we might assume we know the applied forces, the initial and deformed configuration, and the goal is to determine the model parameters that minimize some distance (metric) between initial and deformed configurations. In an inverse deformation analysis, we assume we know the applied forces, boundary conditions, model parameters, and the deformed configuration. The objective is to determine the undeformed configuration that would lead to the deformed configuration if the external forces were to be applied.

1.3 Background

Several authors have tackled the problem of inverse deformation analysis using a variety of strategies. To the best of our knowledge, Adkins [7] was the first to propose exchanging the role of the deformed and undeformed configurations, i.e., to express the displacement of the body as a function of the deformed state. The study was limited to plane strain deformations and uniform extension. Schield [8] applied the same formalism to a homogeneous elastic material, without body force. He showed the equivalence of the equilibrium equations if the initial and deformed configurations are interchanged as well as the volumetric strain energies. The results provided by this approach are shown to be commensurate with those of Adkins [7] but are based on dual relations between the initial configuration and the deformed configuration. Carlson [9] used a variational principle to achieve the same as Schield [8] and showed the validity of the approach for different elastic materials. More recently, Carroll [10] mathematically analyzed the Schield transformation and the proven inverse deformation theorem. The theorem states that if a particular deformation is supported without body force for a specific strain energy W, then the inverse deformation is another energy W^*, derived from the first: $W^*(\boldsymbol{F}) = \det(\boldsymbol{F})W\left(\boldsymbol{F}^{-1}\right)$, where $\boldsymbol{F}$ is the deformation gradient.

Govindjee [11, 12] introduced the reparameterization of the weak form of the forward problem of finite elasticity as a solution method for the inverse problem. This approach only requires C_0 continuity and has a direct physical connection to the problem. In addition, the procedure eliminates boundary condition difficulties, can be straightforwardly implemented using standard forward numerical methods, and can deal with both compressible or incompressible materials.

Inspired by Govindjee [12] (Eulerian model) and Yamada [13] (arbitrary Lagrangian–Eulerian (ALE)), Fachinotti [14] rewrote the constitutive equations in terms of Lagrangian variables. In contrast to Eulerian and ALE variables, the Lagrangian formulation requires only a few modifications from the direct to the inverse analysis code (i.e., the computations of the finite-element residual vector and the Jacobian matrix). The formulation is convenient and allows to solve *inverse design problems* such as finding the unloaded shape of a turbine blade under known loading. However, few drawbacks arise such as the difficulty of deriving and implementing the consistent Jacobian of the finite-element formulation. Despite the usefulness of the approach, to our knowledge, this type of analysis is still not available in any widely used commercial simulation software.

Iterative methods identify the undeformed configuration based on several forward calculations. The algorithm is introduced by Sellier [15] with a fixed-point method for elastostatic problems and then generalizes as the *backward displacement method* by Bols [16] for patient-specific blood vessel simulations. The iterative algorithm of Sellier has been widely applied to many image-based biomechanical simulations, mainly thanks to its algorithmic simplicity and its ability to use a standard non-linear elasticity simulation software [4]. However, when applied to strongly non-linear problems resulting from material or geometric non-linearity, the algorithm lacks robustness. Furthermore, iterative methods usually require at least one non-linear elasticity problem solution, resulting in higher costs compared with the approach of Fachinotti [14].

In the computer graphics community, Chen [17] used asymptotic numerical methods (ANM) to compute the rest-shape of elastic objects with a neo-Hookean material model. The ANM considers a parametrized version of the static equilibrium: $f(x, X) + \lambda g = 0$, where g is gravity, λ a loading parameter and f are the internal forces with the given deformed configuration x and the unknown rest-configuration X. Then, the algorithm incrementally computes the asymptotic expansion of the curve in (X, λ) space until $\lambda = 1$, which corresponds to the rest-position. In this study, ANM offers superior performance, robustness, and convergence speed over traditional Newton-type methods for highly nonlinear material models. However, the major drawback of the method is the complexity of changing the model formulation. Indeed, using a different material model implies to establish a different quadratic relationship between Cauchy stresses and the rest-position, then deriving the asymptotic local expansion. More recently, Ly [18] developed an inversion algorithm applicable to geometrically non-linear thin shells, including the effects of contact and dry friction with an external body.

1.4 Contribution

In this paper, we propose to use the Lagrangian formulation of Fachinotti [14] coupled with automatic code generation tools provided by the FEniCS Project finite-element software [19] to compute the rest or undeformed configuration of an object knowing the deformed configuration, the external loads and the material properties. We show experimental validation that the methodology is effective at recovering the undeformed configuration from imaging data. The formulation requires only a few minor modifications of the direct simulations, making it easy to implement. The automated differentiation tools from FEniCS Project provide a great deal of flexibility, for example, permitting users to quickly and easily modify the material model to suit their own problem.

1.5 Outline

This paper is organized as follows; first, we explain the inverse deformation analysis methods as well as the near-incompressible hyperelastic model used for the study. We test our formulation on some simple analytical cases described in [20, 21]. Then, we show in a numerical example how our variational formulation can surpass the iterative algorithm proposed by Sellier [15]. Finally, we demonstrate a relevant real-world application by retrieving the undeformed configuration of a Polydimethylsiloxone (PDMS) beam under the action of gravity from imaging data.

2 Inverse finite strain elasticity formulation

This section presents two methods to compute the undeformed configuration knowing the deformed configuration under known loading. We first introduce our methodology derived from Fachinotti [14], then we briefly outline a simple iterative geometric algorithm described in [15].

2.1 Inverse FEM method

2.1.1 Mathematical operators

In this paper, we define the colon operator (:) as the inner product between tensors (summed pairwise product of all elements). Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be rank-2 tensors, the inner product is then defined as $\boldsymbol{A} : \boldsymbol{B} = \mathrm{tr}(\boldsymbol{A}^T\boldsymbol{B}) = A_{ij}B_{ij}$ using the Einstein notation. In a same way, we define the dot product between vectors as $\boldsymbol{u} \cdot \boldsymbol{v} = u_i v_i$. Finally, $\nabla(\bullet)$ denotes the vector differential operator such that $\nabla \boldsymbol{v} = \frac{\partial v_j}{\partial x_i}$.

Consider a deformable body $\mathcal{B}$. We denote the undeformed configuration Ω_0. The location of a particle of $\mathcal{B}$ in Ω_0 is denoted $\boldsymbol{X}$. Conversely, the deformed configuration is noted Ω, and the location of a particle of $\mathcal{B}$ in Ω is noted $\mathbf{x}$. A one-to-one mapping $\boldsymbol{\phi}$ maps the position of a particle $\boldsymbol{X}$ in Ω_0 to the position of the same particle $\boldsymbol{x}$ in Ω, i.e., $\boldsymbol{x} = \phi(\boldsymbol{X})$. The configuration Ω can be obtained by $\boldsymbol{\phi}\left(\Omega_0\right) = \left\{\boldsymbol{\phi}(X) \mid X \in \Omega_0\right\}$.

The weak form of the static equilibrium of the inverse deformation is expressed in the deformed configuration:

$$\int_{\Omega} \boldsymbol{\sigma} : \nabla \boldsymbol{\eta} \, \mathrm{d}\Omega = \int_{\Omega} \rho \boldsymbol{b} \cdot \boldsymbol{\eta} d\Omega + \int_{\Gamma} \boldsymbol{t} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma, \tag{1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\nabla(\bullet)$ the gradient in Ω with respect to the deformed spatial position of the test function $\boldsymbol{\eta}$. ρ is the density of the material in the deformed configuration and $\boldsymbol{b}$ are the external forces in the deformed configuration. Finally, a traction boundary condition prescribed on a part Γ of the boundary as $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}$ with $\boldsymbol{n}$ the outward unit normal at the boundary. On the remaining part of the boundary, we assume that the value of the displacement is given, i.e., a Dirichlet condition.

This approach has the advantage of being based on classical mechanical principles. However, mechanical quantities such as strains or stresses are defined depending on $\boldsymbol{X}$. Whereas, in the inverse approach, the initial geometry is replaced by the deformed geometry $\boldsymbol{x}$. We redefine the classical displacement $\boldsymbol{u}(\boldsymbol{X}) = \boldsymbol{x} - \boldsymbol{X} = \boldsymbol{\phi}(\boldsymbol{X}) - \boldsymbol{X}$ as

$$\boldsymbol{u}'(\boldsymbol{x}) = \boldsymbol{X} - \boldsymbol{x}. \tag{2}$$

Note that trivially

$$\boldsymbol{u}' \circ \boldsymbol{\phi} + \boldsymbol{u} = 0. \tag{3}$$

This redefinition does not modify the classical finite-element pipeline: the unknown position is still the first term in which the known position is subtracted. In addition, as we compute gradients in the deformed configuration, this necessitates the redefinition of the deformation gradient as well:

$$\begin{aligned} \boldsymbol{F} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X})}{\partial \boldsymbol{X}} &= \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{x}}\right)^{-1} = \left(\frac{\partial \boldsymbol{u}'(\boldsymbol{x})}{\partial \boldsymbol{x}} + \boldsymbol{I}\right)^{-1} \\ &= \left(\nabla \boldsymbol{u}' + \boldsymbol{I}\right)^{-1}. \end{aligned} \tag{4}$$

Henceforth, when performing an inverse deformation analysis, $\boldsymbol{F}$ and all derived quantities (strain measures, invariants, energy densities, stress measures etc.) are always computed using the above redefinition in terms of $\boldsymbol{u}'$.

This new formulation requires us to only make one change compared to the classical direct finite-element pipeline; rewrite $\boldsymbol{F}$ in terms of $\boldsymbol{u}'$. This formulation can find the undeformed configuration of an object, knowing only the deformed configuration and the applied forces. The process is not iterative (we perform several Newton–Raphson iterations but only one simulation) and based on the equation of continuum mechanics.

2.2 Nearly-incompressible constitutive model

For many materials, simple elastic models such as the St. Venant Kirchhoff model are not sufficient to describe the observed behavior. More complex hyperelastic models provide a mechanism of modeling the stress–strain behavior of complex materials, such as elastomers or biological tissues. They can be assumed compressible, i.e., the volume may change during deformation or nearly incompressible or even completely incompressible, i.e., the volume is preserved during deformation det$\boldsymbol{F} \sim 1$.

For a hyperelastic material, the strain energy density function describes the stored energy as a function of the isochoric deformation, i.e., shape deformations without volume change. However, using the standard displacement-based finite-element method to describe incompressible material behavior may cause numerical problems typically referred to as *locking*. Simply put, locking occurs when too many constraints are imposed on the discrete formulation and its overall approximation power is destroyed.

To overcome these difficulties, mixed formulations have been developed. In these formulations, the variational principle is modified by writing the potential energy functional. The strain energy is expressed in terms of the deviatoric component only and the incompressibility constraint is explicitly enforced using a Lagrange multiplier with physical meaning akin to pressure (p). It turns out that the Lagrange multipliers can be expressed as a function of the hydrostatic pressure values $f(p)$ [34]:

$$\psi(\boldsymbol{C},p) = \psi_{\text{deviatoric}}(\boldsymbol{C}) + f(p), \tag{5}$$

$$\begin{aligned} f(p) =& \psi_{\text{hydrostatic}}(J), \text{ with } p = \sigma_{\text{hydro}} \\ =& \frac{1}{3}\text{tr}(\boldsymbol{\sigma}) = \frac{\partial \psi_{\text{hydrostatic}}(J)}{\partial J}, \end{aligned} \tag{6}$$

where J is the Jacobian defined as $J = \det \boldsymbol{F}$ and $\boldsymbol{C}$ is the right Cauchy–Green deformation tensor. Among the hyperelastic materials, we chose to use neo-Hooke [33] and Mooney–Rivlin [31, 32] models, often used for modeling soft object deformations.

Neo-Hookean By calculating $f(p)$, we can deduce the mixed displacement–pressure formulation of a nearly-incompressible neo-Hookean material:

$$\psi_{\text{NH}}(\boldsymbol{C},p) = \frac{\mu}{2}(\text{I}_B - 3) - \mu \ln(J) + p\ln(J) - \frac{1}{2\lambda}p^2, \tag{7}$$

with

$$\psi_{\text{deviatoric}}(\boldsymbol{C}) = \frac{\mu}{2}(\text{I}_B - 3) - \mu \ln(J), \tag{8}$$

and

$$\psi_{\text{hydrostatic}}(J) = \frac{\lambda}{2}\ln(J)^2, \tag{9}$$

where λ and μ are material constants called the Lamé parameters.

Mooney–Rivlin By calculating $f(p)$, we can deduce the mixed displacement–pressure formulation of a nearly-incompressible Mooney–Rivlin material:

$$\psi_{\text{MR}}(\boldsymbol{C},p) = C_1\left(\overline{\text{I}_C} - 3\right) + C_2\left(\overline{\text{II}_C} - 3\right) + p(J-1) - \frac{1}{4D_1}p^2, \tag{10}$$

with C_1, C_2, D_1 material constants, in addition of

$$\psi_{\text{deviatoric}}(\boldsymbol{C}) = C_1\left(\overline{\text{I}_C} - 3\right) + C_2\left(\overline{\text{II}_C} - 3\right), \tag{11}$$

and

$$\psi_{\text{hydrostatic}}(J) = D_1(J-1)^2, \tag{12}$$

with the modified invariants $\overline{\text{I}_C} = J^{-\frac{2}{3}}\ \text{I}_C$, $\overline{\text{II}_C} = J^{-\frac{4}{3}}\ \text{II}_C$ and classic invariants $\text{I}_C = \text{tr}(\mathbf{C})$, $\text{II}_C = \frac{1}{2}\left((\text{tr}(\mathbf{C}))^2 - \text{tr}\left(\mathbf{C}^2\right)\right)$. These nearly-incompressible energy densities are used to generate the results from the FEniCS Project presented in this paper.

Finally, the strain energy density can be related to the second Piola–Kirchhoff $\boldsymbol{S}$ and then to the Cauchy stress tensor by recalling the relation $\frac{\partial J}{\partial \boldsymbol{C}} = \frac{1}{2}J\boldsymbol{C}^{-1}$:

$$\boldsymbol{S} = 2\frac{\partial \psi}{\partial \boldsymbol{C}} = 2\left(f\prime(p)\frac{\partial J}{\partial \boldsymbol{C}} + \frac{\psi_{\text{deviatoric}}}{\partial \boldsymbol{C}}\right), \tag{13}$$

$$\boldsymbol{\sigma} = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T J = f\prime(p) + 2J^{-1}\boldsymbol{F}\frac{\psi_{\text{deviatoric}}}{\partial \boldsymbol{C}}\boldsymbol{F}^T, \tag{14}$$

where $f\prime$ denotes the differentiation of $f(p)$ with respect to p.

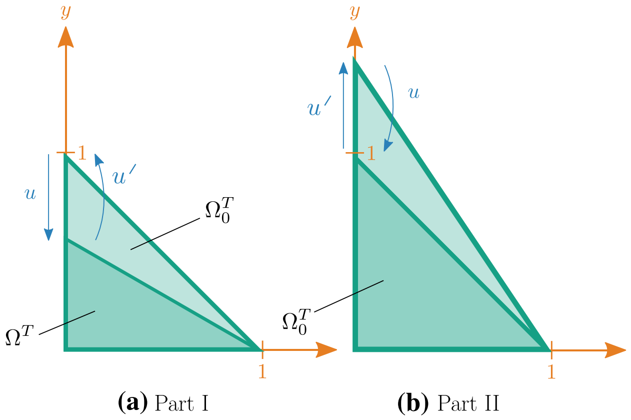
2.3 Finite-element solver

We use the FEniCS Project finite-element software [19] to discretise both the standard finite strain elasticity problem and the inverse finite strain elasticity problem that we will outline in the next section. We use a mixed displacement–pressure finite-element formulation with second-order continuous Lagrangian finite elements for displacement $\mathbf{u}$ and first-order continuous Lagrangian finite elements for pressure p. This pairing is well-known to be inf-sup stable [36] and relatively robust with respect to numerical locking.

The variational form of the residual Eq. 1 is defined in the unified form language (UFL) [22] and symbolically differentiated to derive an expression for consistent Jacobian. The FEniCS form compiler (FFC) [23] is used to automatically generate low-level C++ code from the high-level UFL description that can calculate the Jacobian and residual cell tensors. The overall solution process is driven by the DOLFIN finite-element library [24]. We use a standard Newton–Raphson algorithm with continuation in the loading parameter. The linear system within the Newton–Raphson algorithm is solved using the direct solver MUMPS via PETSc [25]. To be more precise, inside PETSc we use the direct sparse MUMPS solver as a preconditioner to a single iteration of a Krylov method, leading to convergence in one iteration. The relative tolerance of the solve is in the order of $1e^{-12}$. The complete implementation of the standard or inverse problem is around 100 lines of Python code that closely follows the mathematical structure of the problem. We refer the reader to the supplementary material [26] for further details.

2.4 Iterative geometric algorithm

Sellier [15] proposed an iterative geometric algorithm (IGA, not to be confused with isogeometric analysis). The algorithm is simple to implement and only requires an existing (standard) forward deformation solver. The algorithm starts with an initial guess for the undeformed configuration (usually chosen, for lack of a better choice, the deformed one) and applies successive displacement fields to it until a convergence criterion is reached. The sequence of displacement fields is obtained from the direct simulations of the current rest-configuration undergoing external forces. The shape of the object after the direct simulation provides an error compared to the exact rest-configuration by measuring the distance to the initial configuration. An updated estimate of the undeformed configuration is calculated by correcting the previous guess with the difference between the computed and deformed configuration. The algorithm stops when the error (computed using the l^2-norm) is below a defined threshold ϵ or a maximum number of iterations $NB^{\max}$ has been reached. The process is outlined in Algorithm 1.



(a) Part I

(b) Part II

Fig. 2 Single tetrahedron test, plane view. The two bottom nodes are fixed ($\boldsymbol{u} = (0, 0, 0)^T$ for $y = 0$). In part I, a force $\boldsymbol{f}$ is applied on the "free node" ($y = 1$) generating a displacement $\boldsymbol{u}$. Then a second force $\boldsymbol{f'} = \boldsymbol{f}$ (generating $\boldsymbol{u'}$) is applied on the same node to retrieve the initial configuration. In part II, we reiterate the same procedure as part I except that we first apply $\boldsymbol{f'}$ and then $\boldsymbol{f}$ on the "free node"

Algorithm 1: Iterative geometric algorithm from Sellier [15].

```
$X^0 \leftarrow X^{\text{ini}}$
run direct simulation 0 with $X^0$ the initial configuration
$u^0 \leftarrow x^0 - X^0$
err $\leftarrow$ error between $x^0$ and $X^{\text{ini}}$
$j \leftarrow 1$
while err $> \epsilon$ and $j < NB^{\max}$ do
    $X^j \leftarrow X^{(j-1)} - u^{(j-1)}$
    run direct simulation j with $X^{(j-1)}$ the initial configuration
    $u^j \leftarrow x^j - X^j$
    err $\leftarrow$ error between $x^j$ and $X^{\text{ini}}$
    $j \leftarrow j + 1$
end
```

3 Numerical results

The inverse deformation framework is very similar to the traditional direct framework. To assess the numerical precision of the inverse method, we first applied a serie of tests to verify the soundness of the direct and inverse approaches in which an analytical solution is known. We tested our formulation on some simple analytical cases described in [20, 21], such as simple and generalized shears. Due to the triviality of the tests, they are detailed in the Appendix and can be found in the supplementary material [26].

3.1 Single unit tetrahedron

3.1.1 Part I

Let us consider a mesh with a single unit tetrahedron with a linear Lagrangian finite-element space. Its domain is denoted Ω_0^T. The nodal coordinates are $[0, 0, 0]^T$, $[1, 0, 0]^T$, $[0, 1, 0]^T$ and $[0, 0, 1]^T$. The nodes with $y = 0$ are fixed, leaving only one free node. A uniform force $\boldsymbol{f}$ is applied along the y-axis. The tetrahedron is deformed so that the free node moves along the y-axis.

Table 1 Benchmark results on a single tetrahedron simulation

	Part I			Part II		
	FEM	IGA (1)	IGA (2)	FEM	IGA (1)	IGA (2)
Average error	4.49E−12	2.12E−6	2.28E−12	5.22E−12	1.98E−6	2.28E−12
SD	1.07E−11	1.09E−6	6.69E−12	1.25E−11	1.05E−6	2.41E−12
Minimum	9.26E−22	5.44E−8	4.15E−35	1.04E−21	5.47E−8	6.76E−12
Maximum	5.52E−11	4.48E−6	5.11E−11	7.26E−11	3.99E−6	5.04E−11
Avg #simulations	1	4.70	11.2	1	4.84	13.4
Avg time (ms)	34	162	387	33	162	387
Avg time ratio	1	4.75	11.35	1	4.70	11.70

We compare our inverse FEM deformation algorithm (FEM), with the iterative geometric algorithm (IGA (1)) with the arbitrary convergence criterion 10^{-6}, and with the IGA at the same accuracy than PB (IGA (2)). We compute statistical indicators such as the average error, the standard deviation (SD) of the error, the minimum and maximum errors. In addition, we calculate the average number of simulations (avg #simulations), the average time needed to reach the convergence criterion (avg time) and the average time ratio (avg time ratio) obtained with $\frac{\text{avg time}}{\text{FEM avg time}}$

In a first step, we compute the deformation $\boldsymbol{\phi}$ with the direct method. A displacement $\boldsymbol{u}$ is computed for the free node. The deformed domain is $\Omega^T = \boldsymbol{\phi}(\Omega_0^T)$. In a second step, the initial geometry is the deformed geometry Ω^T, i.e., a unit tetrahedron with the nodes $y = 0$ fixed, and the remaining node displaced from $\boldsymbol{u}$. The same uniform force $\boldsymbol{f'} = \boldsymbol{f}$ is applied. An inverse simulation is computed so that the displacement of the free node is $\boldsymbol{u'}$. This example is depicted in Fig. 2a.

3.1.2 Part II

We consider the same unit tetrahedron, with the same boundary conditions. A uniform force $\boldsymbol{f'}$ is applied along the y-axis.

In the first step, an inverse simulation is computed, leading to a displacement of $\boldsymbol{u'}$. In the second step, the resulting geometry is deformed with a direct simulation leading to a displacement of $\boldsymbol{u}$. This part of the example is depicted in Fig. 2b.

The difference with the first part of the test is the order of the successive simulations. In part I, the inverse simulation is performed after the direct simulation. In part II, it is the opposite. In both parts of the test, the goal is to verify that the following relationship: $\boldsymbol{u'} = -\boldsymbol{u}$.

Furthermore, the inverse simulation is computed with IGA to compare the results and performance with our method. In this test, the error measure is defined as: $||\boldsymbol{u'} + \boldsymbol{u}||_{l^2}$. We measured this error with different constitutive equations and varying their associated mechanical parameters. In total, we performed 153 tests and provided a statistical analysis in Table 1.

We observe that the accuracy of the iterative algorithm depends on the number of iterations, but it also increases the computational cost, because each iteration calls a direct simulation. Our method provides high accuracy while requiring only the solution of a problem with similar complexity to a single iteration of IGA. Beyond the numerical results, one point is that in 7 tests over the 153 of the part II, the iterative algorithm was not able to reach the accuracy of our method within 50 iterations.

3.2 Sagging block

In Appendix, we applied different displacements on a cube using simple and generalized shears. For the sagging block case, we use the same pipeline as mentioned in Sect. 3.1. Namely, we deform and object by applying a force and recover its initial shape by exercising the opposite force on the deformed object. The geometry is a three-dimensional unit cube of 1 m length, fixed at the bottom (Dirichlet condition $\boldsymbol{u} = (0, 0, 0)^T$ when $y = 0$) with gravity as a body force applied to the entire object. For the material model, we used the incompressible Neo-Hookean defined in Eq. 7. We performed several Monte Carlo (MC) simulations which to study the error of the recovered initial shape using different material parameters. For the sake of simplicity, we assumed that λ and μ were both following an uniform distribution between an acceptable range of values:

$$\mu \sim U(2.0 \times 10^3, 2.0 \times 10^5),$$
$$\lambda \sim U(8.0 \times 10^4, 8.0 \times 10^6).$$

We empirically chose those values as lower and upper limits, because we could observe large visible deformations within these bounds. We draw 100 samples of each parameters and make all possible combinations leading to $100^2 = 10{,}000$ simulations. To study the error, we calculated the mean absolute error (MAE) between the initial shape and the initial shape predicted by our inverse deformation

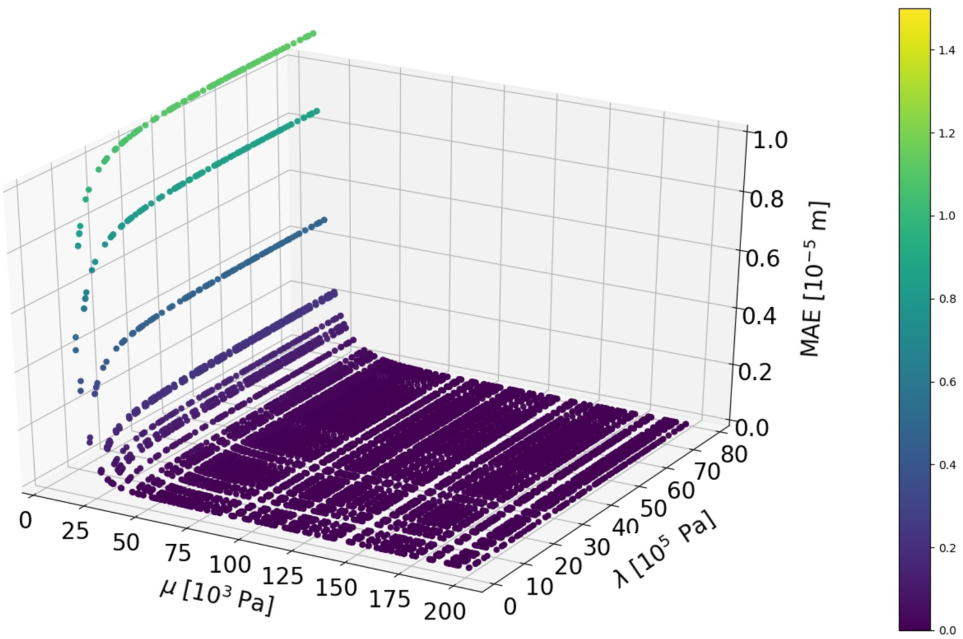


Fig. 3 Sagging block Monte Carlo simulations using a uniform distribution of Neo-Hookean material parameters. The mean absolute error (MAE) between the initial shape and the predicted shape of the inverse deformation algorithm is calculated for 10,000 simulations

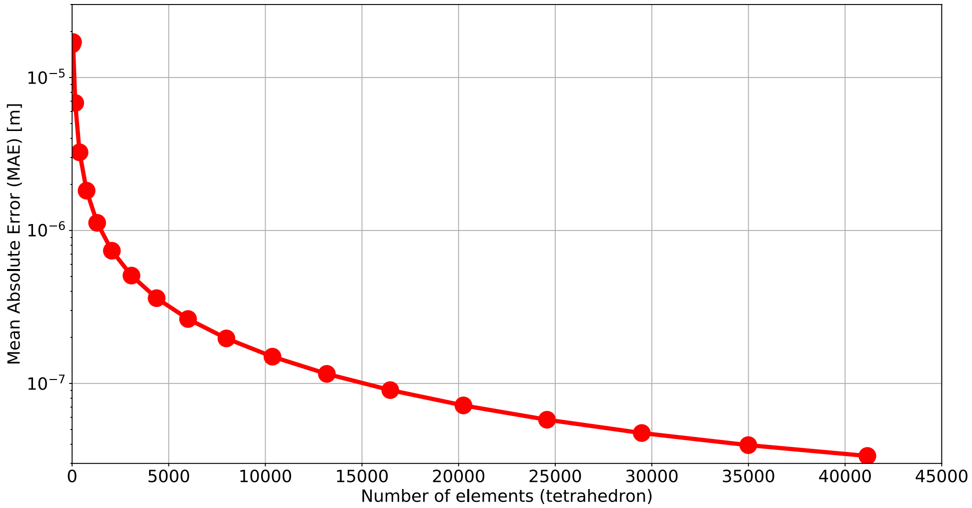


Fig. 4 Semilog plot of the mesh convergence analysis of the sagging block simulation. The mean absolute error (MAE) between the initial shape and the predicted shape of the inverse deformation algorithm is calculated for different mesh refinement with fixed parameters ($\lambda = 80 \times 10^4$ Pa and $\mu = 25 \times 10^3$)

algorithm. The MAE is defined as $\mathrm{MAE} = \frac{\sum_i^N |e|}{N}$, where $e = x_{\text{prediction}} - x_{\text{initial}}$ and N the number of points of the cube.

The result of the MC simulations are shown in the 3D plot in Fig. 3. We observe a small error when the material parameters are on the upper limit of our distribution, e.g., a $\mathrm{MAE} = 1.21 \times 10^{-9}$ m for $\mu = 175 \times 10^3$ Pa and $\lambda = 50 \times 10^5$ Pa. On the contrary, when moving to smaller values, e.g., $\mu = 20 \times 10^3$ Pa and $\lambda = 30 \times 10^5$ Pa, the MAE is increasing to 1.21×10^{-5} m. This behavior is expected as choosing smaller values of λ and μ induce larger displacements increasing the MAE. Hence, the obtained MAE are more than acceptable and choosing lower values of the material parameters will surely increase the MAE but also create convergence issues for such large deformations. Furthermore, we performed a mesh convergence analysis shown in Fig. 4 proving that refining the mesh leads to a smaller MAE.

4 Experimental results

In this section, we will demonstrate that our inverse simulation method can match the outcome of a real experiment and, therefore, has value as a predictive modelling tool.

We fixed one extremity of a beam made from Polydimethylsiloxane (PDMS) to a vertical support and allowed it to deform under gravity, slowly accompanied to the equilibrium position by hand and released in that position. The deformed steady state was reached in about 1 min (after

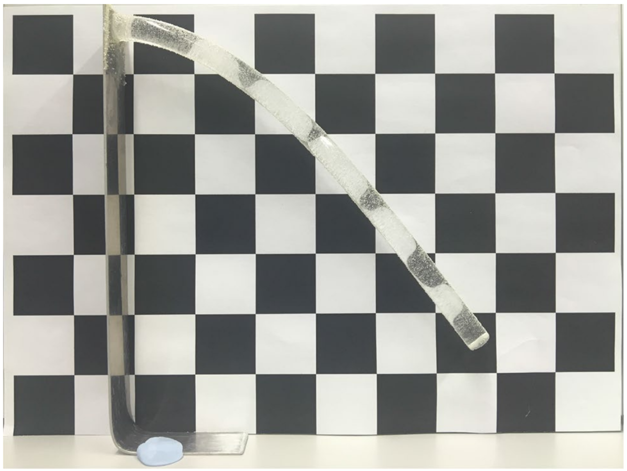


Fig. 5 Experimental setup: initially a straight PDMS cylindrical beam of 182 mm length and 8.5 mm diameter, clamped on the left side and deformed by gravity

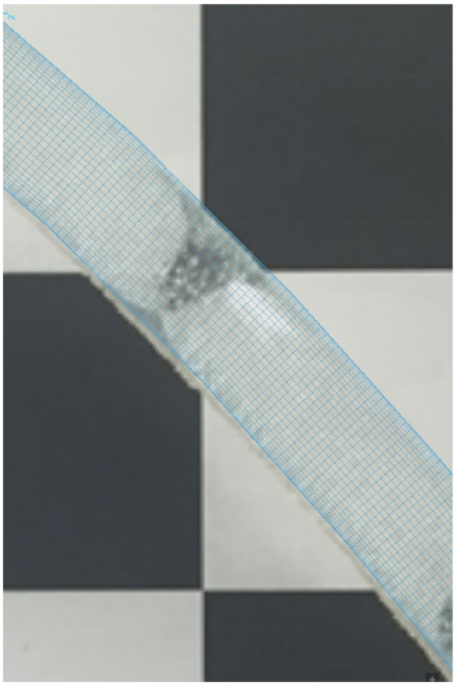


Fig. 6 Manual process in Blender to delineate the contours and extract the 3D mesh of the deformed configuration

vibrations were completely damped) resulting in the configuration displayed in Fig. 5. To conduct the experiment, the room environmental temperature was recorded to 19°C and the humidity at around 50% (the metrology lab is set to keep these temperatures and humidities). It is known that the mechanical properties of the PDMS material are very sensitive to its curing temperature. On the contrary, the material is very stable once it is fully cured and the mechanical properties are constant for a large range of temperatures centered around the 19°C used for testing. This is obviously not the case if the sample is heated up to temperatures in the range of 80–90°C, but these are not biological values and are out of the scope of the study. Note that the humidity effects are negligible, since the material is quite hydrophobic. To extract the mesh of the deformed configuration from the image, we used the software Blender[1] and contoured the beam on 2D images by hand, as shown in Fig. 6. From Blender, we exported a quadrilateral surface mesh as an.obj file. Then, using the CGAL library, we generated tetrahedrons from the surface mesh. A custom code converted the tetrahedra data structure to a dolfin-compatible.xml file. This code is available as a submodule on our Github page [26]. This mesh will be called the "reference" and used as ground-truth for this section.

To run the inverse deformation algorithm, we need three input parameters: the applied force field, the deformed configuration, and the mechanical properties. In this section, the force field is gravity and the deformed configuration was obtained by manual processing. A separate experiment was performed to obtain the mechanical properties and will be detailed in the following section.

4.1 Materials

We used a PDMS (Sylgard 184, Ellsworth Adhesives) cylinder of density $965\mathrm{kg}/\mathrm{m}^3$ of undeformed dimensions 182 mm and 8.5 mm for length and diameter, respectively.

For the sample preparation the elastomeric part and curing agent were mixed in a 10:1 ratio and cured at room temperature for 24 h before being tested [27]. A surgical knife was used for cutting cylindrical shapes from the second cylinder of PDMS, for compression tests samples (diameter 11 mm, height 7 ± 1 mm).

To characterize the material properties, we used the Mach-1$^{\mathrm{TM}}$ mechanical testing system (Biomomentum, Canada) as a testing rig for the unconfined compression tests. We used the following protocol:

- A 1.5 mm single-axis load cell with a resolution of 75μN was used to measure the vertical force.
- The vertical displacement was measured by the moving stage of the rig with a resolution of 0.1μN.
- To minimize friction, paraffin oil was used between the sample and the compression platens.
- One loading cycle was executed on each specimen. To detect the response of the material at large strains, the samples were compressed at a constant speed of 0.083 mm/s until a displacement corresponding to 30% of the

[1] https://www.blender.org/.

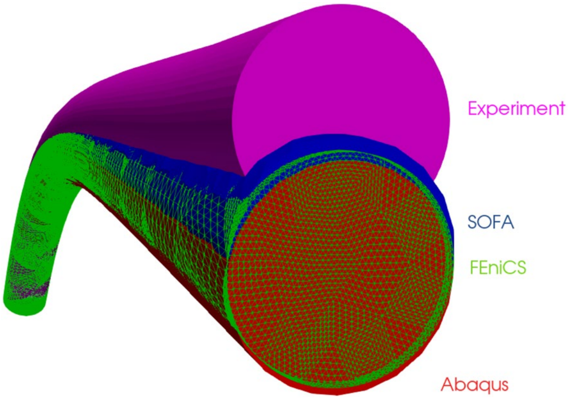


Fig. 7 Comparison of the deformation of the beam for the direct simulation of three different softwares. The three simulations have the same geometry, boundary conditions, material model, and mechanical parameters corresponding to the experiment's material (PDMS) found in Sect. 4.1. From top to down, in magenta: the experimental data, in blue: the SOFA simulation, in wire-frame green: the FEniCS simulation and in red: the Abaqus simulation

measured height was achieved. Particular attention was used to monitor the samples that had uniformly expanded in the radial direction and that their upper and lower faces remained adhered to the moving platen and the fixed platform for the entire duration of the test.

- The Abaqus evaluation routine was used to fit the true stress–true strain experimental curves with a Mooney–Rivlin model. Abaqus employs a linear least-squares fit for the Mooney–Rivlin form to find the optimal model parameters.

In our case, the optimal parameters are: $D_1 = 7.965 \times 10^{-8}$ Pa, $C_{10} = 101.709$ kPa, $C_{01} = 151.065$ kPa. The value of D_1 is close to 0 which validates our incompressibility hypothesis. The two obtained values of C_{10} and C_{01} are the optimal values to describe the behavior of the PDMS using a Mooney–Rivlin material formulation.

4.2 Direct simulation

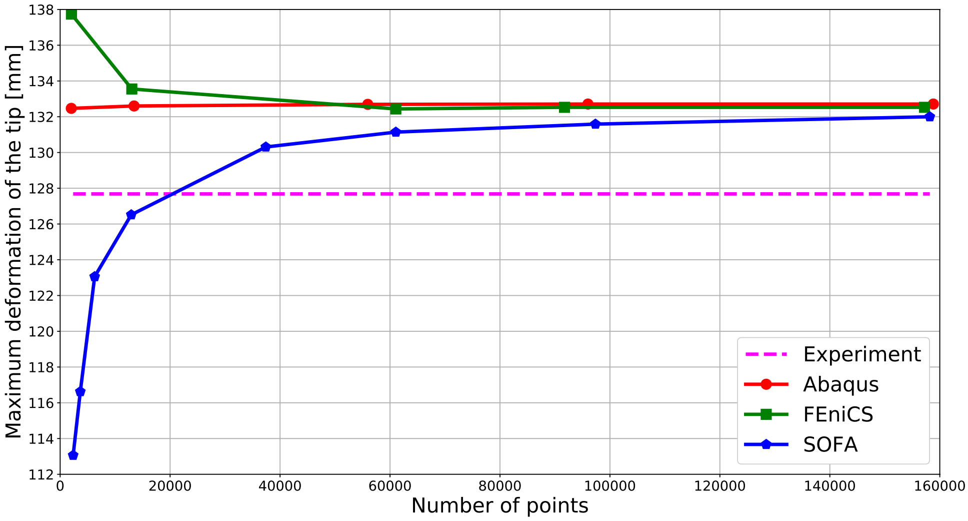
We run three simulations with identical geometry, material properties, boundary conditions, and material model to verify that these commonly used softwares produce quantitatively similar results. For each software, we used an incompressible Mooney–Rivlin model and boundary conditions imitating the setup shown in Fig. 5. Namely, a Dirichlet boundary condition of $\boldsymbol{u} = 0$ on the left side of the beam (to imitate the clamping of the beam to the support) and the gravity is applied as a body force on the entire object.)

FEniCS We use the same model as described in Sect. 2.2.

Abaqus We use a static step with a gravity load to solve the beam deformation in Abaqus. Abaqus/Standard uses Newton's method as a numerical technique for solving the nonlinear equilibrium equations. We employed C3D8RH elements, an 8-node linear brick, hybrid/mixed, constant pressure, reduced integration with hourglass control. Hybrid elements are usually used in Abaqus when the material definition is close to incompressibility to avoid locking. The reduced integration is used to speed up computational time and avoid numerical locking.

SOFA We employ the *multiplicative Jacobian energy decomposition* method (MJED) which is an optimized algorithm for building the stiffness and tangent stiffness matrices of non-linear hyperelastic materials [28]. An MJED implementation is available in SOFA [29] for finite- element formulation using linear tetrahedral elements. The linear system of equations is solved in every step of quasi-static simulation

Fig. 8 Mesh convergence analysis of the forward simulation. We calculate the maximum deformation of the tip of beam for several level of refinement of the mesh



using a fast in-house linear equation solver based on the Cholesky [35] decomposition.

The visual result of the forward simulations using the 3 softwares are shown in Fig. 7. For each model, we perform a mesh convergence analysis shown in Fig. 8, where we plot the maximum deformation of the beam (located at the tip) for different mesh resolutions.

We observe that the tip displacement for the three software converges to similar solutions (FEniCS: 132.52 mm, Abaqus: 132.71 mm, SOFA: 130.31 mm), while the experimental value is 127.68 mm. We observe a small difference between the numerical solutions and the experiment.

FEniCS and Abaqus give similar results, while SOFA is 2 mm off, compared to the 2 others softwares. We observe in Fig. 8 that FEniCS and Abaqus converged with 60,000 points, while SOFA is still not converged with 160,000 points (due to computational resources limitations). One reason is that SOFA is usually designed for real-time simulation and only uses dynamic solvers which can lead to inaccuracy compared with static solvers from FEniCS and Abaqus. Furthermore, the differences between numerical solutions can be explained by the use of three slightly different formulations of the Mooney–Rivlin law as well as different solvers for solving the equation. Finally, one can note that the ABAQUS simulation converges very quickly. Possibly because ABAQUS is using a hexahedral cell shape with quadratic polynomial interpolation of the displacement field, against tetrahedron cells in FEniCS and SOFA. It is well known, at least in the Engineering community, that finite elements based on hexahedral cells often perform better in practice.

Some factors can explain the difference between the numerical solutions and the experimental value. For instance, the variation may be explained by inadequate constitutive equations or boundary conditions. Then, uncertainties in the mechanical properties measures may also be a factor, especially because the PDMS might exhibit slightly asymmetric behaviour under compression and tension. Finally, we obtained the reference mesh of the undeformed configuration manually based on 2D imaging data, where inaccuracies can be introduced.

4.3 Inverse simulation

In the previous section, we compared the forward simulations of three different software with our experimental solution. In this section, we want to verify the possibility of retrieving the undeformed configuration of our experimental solution knowing only the surface of the deformed configuration, the known applied loads and the material properties.

For this, we converted our experimental surface mesh of the deformed configuration into a volumetric mesh and applied our inverse deformation algorithm implemented using FEniCS. We previously showed a deformation difference of 4.84 mm for the forward simulation in FEniCS. Of course, we do not expect to obtain a perfectly straight beam (the ideal undeformed configuration), but rather an error on the same order as in the forward simulation (i.e., 4.84 mm).

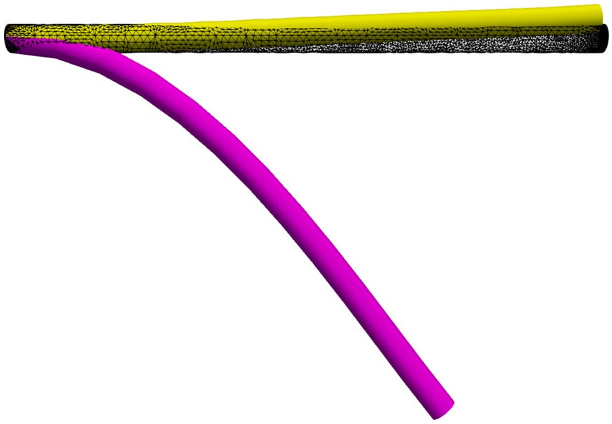


Fig. 9 3D plot of the inverse simulation. From down to top, in magenta: the experimental data we wish to retrieve the undeformed configuration, in wire-frame black: the theoretical straight beam, in yellow: the result of the FEniCS inverse simulation

We show in Fig. 9 the result of the inverse deformation algorithm. As expected, the inverse simulation (in yellow) applied to the experimental data (deformed configuration in magenta) is slightly different from the theoretical straight beam that we should obtain (in black). To be more precise, we achieve an error of 5.36 mm compared with the idealized straight beam. As mentioned previously, we expect an error on the order of that for the standard deformation problem (4.84 mm) due to the inherent parametric and modeling uncertainties (material model, material properties, boundary conditions, geometry) already discussed. We, therefore, judge that the proposed methodology has strong potential for prediction of the undeformed configuration of a soft body.

5 Conclusions

In the present paper, we performed two physical experiments and one numerical experiment to address the *inverse deformation problem*. First an unconfined experiment to find the material properties, and then the cantilever beam problem to find the deformed configuration. The inverse deformation was a computational exercise to see if the undeformed configuration could be recovered.

Our study used the Lagrangian formulation of [14] as a basis for implementing the inverse algorithm in the FEniCS Project finite-element software. We took advantage of the automatic differentiation and code generation capabilities to bypass the difficulties of deriving and implementing the consistent Jacobian. The user must then supply the deformed configuration, the mechanical properties, boundary conditions and the applied forces. The user can easily modify the input mesh, run the code efficiently in parallel, change the constitutive model or change the boundary conditions according to their needs. We have made the code and data available in the supplementary material.

We applied the approach to simple academic examples, where we considered two different incompressible hyperelastic models (neo-Hookean and Mooney–Rivlin) and different boundary conditions. We demonstrated on a simple test case that our method is more efficient in terms of robustness and accuracy than the IGA method of [15]. We have only compared with the classical IGA method of Sellier but other works like [30] have improved on this algorithm. However, we can say that unless an iterative approach requires only one forward model solution, in most circumstances the mechanics-based approach detailed here is likely to be faster and more robust.

Finally, we applied the method to an experiment with a PDMS beam deformed under gravity. We verified and quantified the performance of the direct simulations of three different widely used software (Abaqus, FEniCS, SOFA). Using the inverse deformation algorithm we achieve an error of 5.36 mm for the tip displacement compared to the idealised straight beam.

Despite our progress in providing a flexible inverse deformation algorithm, some work remains to assess its robustness. Our experiments were only focused on using homogeneous nearly incompressible hyperelastic models. Other works such as [14] were interested in more complex behaviors, such as anisotropy. Similarly, our experiments were only based on simple geometries and more complex geometries should be considered.

We showed the validity of our approach for the beam problem by generating a mesh of the deformed configuration from 2D images and recovering the undeformed configuration. In future work we intend to apply this algorithm to segmented 3D geometries to calculate the undeformed configuration of an organ.

Appendix

A.1 Verification of the direct simulation

The inverse deformation framework is very similar to the traditional direct framework. To assess the numerical precision of the inverse method, we first apply a series of tests to verify the soundness of the direct approach in which an analytic solution is known. The geometry for every example will remain the same with a unit cube discretized in 216 points.

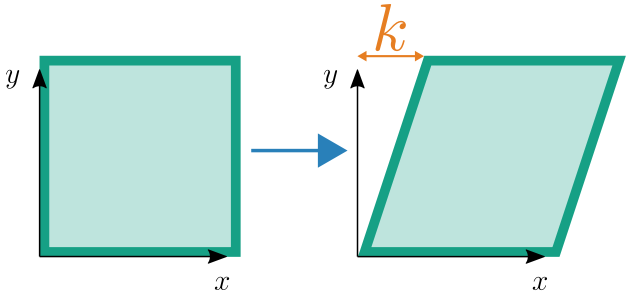


Fig. 10 2D plane cut of a simple shear deformation of a unit cube. An x-displacement of $y \cdot k$ is applied on the boundary, while the bottom is subject to a null Dirichlet boundary ($\boldsymbol{u}_0 = (0, 0, 0)^T$ when $y = 0$)

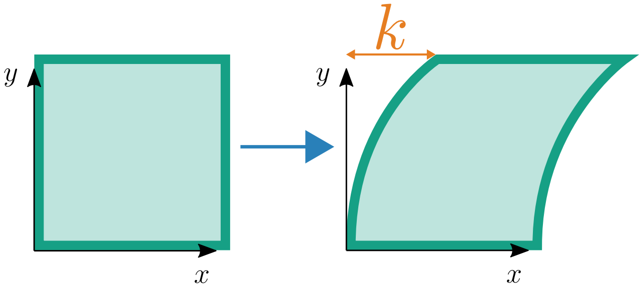


Fig. 11 2D plane cut of a generalized shear deformation of a unit square. An x-displacement of $y^2 \cdot k$ is applied to the boundary of the cube, while the bottom is subject to a null Dirichlet boundary ($\boldsymbol{u}_0 = (0, 0, 0)^T$ when $y = 0$)

A.1.1 Shear deformation

Simple shear Simple shear deformation is a popular benchmark test [20]. The initial geometry is a unit cube with prescribed Dirichlet boundary conditions $\boldsymbol{u}_0 = (y \cdot k, 0, 0)^T$ with y the y-coordinate and k a constant, as illustrated in Fig. 10.

For simple shear deformation, the deformation gradient is equal to

$$\boldsymbol{F} = \nabla_0 \boldsymbol{u} + \boldsymbol{I} = \begin{pmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{15}$$

Now, let us consider a cube made of a Mooney–Rivlin material. Following [20], we can obtain the energy density and the components of the Cauchy stress tensor $\boldsymbol{\sigma}$

$$\psi = k^2(C_1 + C_2), \tag{16}$$

$$
\begin{aligned}
\sigma_{00} &= \frac{k^2(2C_2+4C_1)}{3},\\
\sigma_{11} &= -\frac{k^2(4C_2+2C_1)}{3},\\
\sigma_{22} &= \frac{k^2(2C_2-2C_1)}{3},\\
\sigma_{01} &= k(2C_2+2C_1),\\
\sigma_{02} &= \sigma_{12} = 0.
\end{aligned}
\tag{17}
$$

The values of ψ and $\boldsymbol{\sigma}$ have been evaluated in our framework with several values of k, degrees of discretization, and constitutive parameters. The relative error (by using the l^2 -norm) in strain energy and Cauchy stress tensor, compared to the analytical values, shows the exactness of the direct deformation framework to machine precision (10^{-12} magnitude error).

Generalized shear The generalized shear deformation test is similar to the simple shear deformation [20]. The initial geometry is a unit cube with prescribed Dirichlet boundary conditions $\boldsymbol{u}_0 = (y^2 \cdot k, 0, 0)^T$ with y the y-coordinate and k a constant, as illustrated in Fig. 11.

For generalized shear deformation, the deformation gradient is equal to

$$
\boldsymbol{F} = \begin{pmatrix} 1 & 2ky & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\tag{18}
$$

In the same manner as in the simple shear deformation, we consider a cube made of a Mooney–Rivlin material and can apply the same methods to find the analytical strain energy density function ψ and the Cauchy stress tensor components $\boldsymbol{\sigma}$

$$
\begin{aligned}
\psi &= \int_0^1 [C_1(\overline{I_1}-3) + C_2(\overline{II_1}-3)]\,\mathrm{d}y\\
&= \int_0^1 4k^2y^2(C_1+C_2)\,\mathrm{d}y\\
&= \frac{4k^2(C_1+C_2)}{3},
\end{aligned}
\tag{19}
$$

$$
\begin{aligned}
\sigma_{00} &= \frac{k^2(8C_2+16C_1)}{9},\\
\sigma_{11} &= -\frac{k^2(16C_2+8C_1)}{9},\\
\sigma_{22} &= \frac{k^2(8C_2-8C_1)}{9},\\
\sigma_{01} &= k(2C_2+2C_1),\\
\sigma_{02} &= \sigma_{12} = 0.
\end{aligned}
\tag{20}
$$

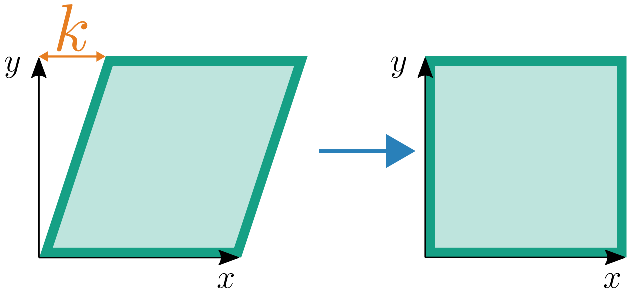


Fig. 12 2D plane cut of an inverse simple shear deformation of a unit cube. An x-displacement of $-y \cdot k$ is applied on the boundary, while the bottom is subject to a null Dirichlet boundary ($\boldsymbol{u}_0 = (0,0,0)^T$ when $y = 0$)

We realize the same tests as the simple shear (different k values, mesh precision, and mechanical parameters) and evaluate the identical quantities, ψ and $\boldsymbol{\sigma}$ values. We observed an impact of the mesh on the strain energy and the Cauchy stress. The error quickly decreases on mesh refinement to reach relative errors under 2%.

A.2 Verification of the inverse simulation

This section presents a series of tests to verify the consistency of our inverse method with the direct approach. More precisely, we show that the undeformed configuration corresponds to the initial configuration used to deform it. During these tests, we also compare our method to the IGA method presented in Sect. 2.4 and evaluate their performance and convergence rates.

A.2.1 Inverse shear deformation

This test is based on the direct shear deformation verification performed in Sect. 1. We verify that the inverse deformation of the simple shear and the generalized shear is consistent with the direct finite-element analysis. The idea is to start the test with the deformed configuration and apply the inverse deformation to verify that the rest-configuration corresponds to the initial geometry of the direct deformation. Since both shear deformations are entirely determined by a displacement field, the inverse deformation consists of applying the opposite displacement field. It is then trivial to claim that the geometry will be recovered, i.e., a unit cube. However, this test also verifies the deformation gradient, the strain energy, and stress tensors are sound. As explained previously, those measures should be equal in both inverse and direct deformation. We verify these statements numerically in these tests.

Inverse simple shear: For the inverse simple shear deformation, the material points are now shifted by $-k \cdot y$ on the x-axis, while the bottom is fixed ($y = 0$). As illustrated in Fig. 12.

We calculate the deformation gradient $\boldsymbol{F}$ which is equal to the deformation gradient in Eq. 15, as expected:

$$\nabla \boldsymbol{u}' + \boldsymbol{I} = \begin{pmatrix} 1 & -k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{21}$$

$$\boldsymbol{F} = (\nabla \boldsymbol{u}' + \boldsymbol{I})^{-1} = \begin{pmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{22}$$

Therefore, the strain energy, which is usually defined depending on $\boldsymbol{F}$, is equal to the strain energy in Eq. 16, and the stress tensor of Eq. 17 remains valid. Since the deformation is homogeneous (constant deformation gradient), our quadratic finite-element method is able to reproduce the analytical solution down to machine precision.

Inverse generalized shear: Similarly, the inverse version of the generalized shear deformation leads to the same deformation gradient tensor (Eq. 18), then to the same strain energy density function (Eq. 19). The relative error is evaluated with different discretizations of the initial mesh but the same parameters set and we obtain with high precision the initial geometry.

Funding This study was supported by European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 764644, No. 798244 and the financial support of the European Research Council Starting Independent Research Grant (ERC StG Grant Agreement No. 279578). Jack S. Hale is supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND) Grant No. 6693582. A.E. Forte acknowledges that this project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 798244. Access to the computing facilities provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CERIT Scientific Cloud LM2015085) was greatly appreciated.

Availability of data and materials The reference [26] (https://doi.org/10.6084/m9.figshare.14035793) contains a full implementations of the forward and inverse deformation problems using the FEniCS Project finite-element software. The latest version is also available on GitHub at https://github.com/Ziemnono/fenics-inverseFEM.

Declarations

Conflict of interest The authors of this work have no conflicts of interest to disclose.

References

1. Raghavan ML, Ma B, Fillinger MF (2006) Non-invasive determination of zero-pressure geometry of arterial aneurysms. Ann Biomed Eng 34:1414–1419. https://doi.org/10.1007/s10439-006-9115-7
2. Lu J, Zhou X, Raghavan ML (2007) Inverse elastostatic stress analysis in pre-deformed biological structures: demonstration using abdominal aortic aneurysms. J Biomech 40:693–696. https://doi.org/10.1016/j.jbiomech.2006.01.015
3. Zhou X, Lu J (2009) Estimation of vascular open configuration using finite element inverse elastostatic method. Eng Comput 25(49):49–59. https://doi.org/10.1007/s00366-008-0104-3
4. Mira A, Carton AK, Muller S, Payan Y (2018) A biomechanical breast model evaluated with respect to MRI data collected in three different positions. Clin Biomech 60:191–199. https://doi.org/10.1016/j.clinbiomech.2018.10.020
5. Koishi M, Govindjee S (2001) Inverse design methodology of a tire. Tire Sci Technol 29:155–170. https://doi.org/10.2346/1.2135236
6. Fachinotti VD, Cardona A, Jetteur P (2008) Finite element modelling of inverse design problems in large deformations anisotropic hyperelasticity. Int J Numer Methods Eng 74:894–910. https://doi.org/10.1002/nme.2193
7. Adkins J (1958) A reciprocal plane property of the finite plan strain equations. J Mech Phys Solid 6:267–275. https://doi.org/10.1016/0022-5096(58)90002-4
8. Schield RT (1967) Inverse deformation results in finite elasticity. Zeitschrift für angewandte Mathematik und Physik ZAMP 18:490–500. https://doi.org/10.1007/BF01601719
9. Carlson DE, Shield T (1969) Inverse deformation results for elastic materials. Zeitschrift für angewandte Mathematik und Physik ZAMP 20:261–263. https://doi.org/10.1007/BF01595564
10. Carroll MM, Rooney FJ (2005) Implications of Shield's inverse deformation theorem for compressible finite elasticity. Zeitschrift für angewandte Mathematik und Physik ZAMP 56:1048–1060. https://doi.org/10.1007/s00033-005-2023-0
11. Govindjee S, Mihalic PA (1996) Computational methods for inverse finite elastostatics. Comput Methods Appl Mech Eng 136:47–57. https://doi.org/10.1016/0045-7825(96)01045-6
12. Govindjee S, Mihalic PA (1998) Computational methods for inverse deformations in quasi-incompressible finite elasticity. Int J Numer Methods Eng 43:821–838. https://doi.org/10.1002/(SICI)1097-0207(19981115)43:5<821::AID-NME453>3.0.CO;2-C
13. Yamada T (1998) Finite element procedure of initial shape determination for hyperelasticity. Struct Eng Mech 6:173–183. https://doi.org/10.12989/sem.1998.6.2.173
14. Albanesi A, Fachinotti V, Cardona A (2009) Design of compliant mechanisms that exactly fit a desired shape. Mecánica Computacional 28:3191–3205

15. Sellier M (2011) An iterative method for the inverse elasto-static problem. J Fluids Struct 27:1461–1470. https://doi.org/10.1016/j.jfluidstructs.2011.08.002
16. Bols J, Degroote J, Trachet B, Verhegghe B, Segers P, Vierendeels J (2013) A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels. J Comput Appl Math 246:10–17. https://doi.org/10.1016/j.cam.2012.10.034
17. Chen X, Zheng C, Xu W, Zhou K (2014) An asymptotic numerical method for inverse elastic shape design. ACM Trans Graph. https://doi.org/10.1145/2601097.2601189
18. Ly M, Casati R, Bertails-Descoubes F, Skouras M, Boissieux L (2018) Inverse elastic shell design with contact and friction. In: SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018, p 37. https://doi.org/10.1145/3272127.3275036
19. Alnæs M, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN (2015) The FEniCS project version 1.5. Archive of Numerical Software 3. https://doi.org/10.11588/ans.2015.100.20553
20. Mihai LA, Goriely A (2013) Numerical simulation of shear and the Poynting effects by the finite element method: an application of the generalised empirical inequalities in non-linear elasticity. Int J Non-Linear Mech 49:1–14. https://doi.org/10.1016/j.ijnonlinmec.2012.09.001
21. Lee CK, Mihai LA, Hale JS, Kerfriden P, Bordas SP (2017) Strain smoothing for compressible and nearly-incompressible finite elasticity. Comput Struct 182:540–555. https://doi.org/10.1016/j.compstruc.2016.05.004
22. Alnæs MS, Logg A, Ølgaard KB, Rognes ME, Wells GN (2014) Unified form language: a domain-specific language for weak formulations of partial differential equations. ACM Trans Math Softw 40:9:1-9:37. https://doi.org/10.1145/2566630
23. Logg A, Ølgaard KB, Rognes ME, Wells GN (2012) FFC: the FEniCS form compiler. In: Logg A, Mardal K-A, Wells G (eds) Automated solution of differential equations by the finite element method, lecture notes in computational science and engineering. Springer, Berlin, pp 227–238
24. Logg A, Wells GN (2010) DOLFIN: automated finite element computing. ACM Trans Math Softw 37:201–2028. https://doi.org/10.1145/1731022.1731030
25. Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp WD, Karpeyev D, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, Zhang H (2019) PETSc Web page. https://www.mcs.anl.gov/petsc
26. Mazier A, Bilger A, Forte AE, Peterlik I, Hale JS, Bordas SPA (2021) Supplementary material for inverse deformation analysis: an experimental and numerical assessment using the FEniCS project. https://doi.org/10.6084/m9.figshare.14035793
27. Forte AE, Galvan S, Manieri F, Rodriguez y Baena F, Dini D (2016) A composite hydrogel for brain tissue phantoms. Mater Des 112:227–238. https://doi.org/10.1016/j.matdes.2016.09.063
28. Marchesseau S, Heimann T, Chatelin S, Willinger R, Delingette H (2010) Fast porous visco-hyperelastic soft tissue model for surgery simulation: application to liver surgery. Prog Biophys Mol Biol 103:185–196. Special Issue on Biomechanical Modelling of Soft Tissue Motion. https://doi.org/10.1016/j.pbiomolbio.2010.09.005
29. Faure F, Duriez C, Delingette H, Allard J, Gilles B, Marchesseau S, Talbot H, Courtecuisse H, Bousquet G, Peterlik I, Cotin S (2012) Sofa: A multi-model framework for interactive physical simulation. In: Soft tissue biomechanical modeling for computer assisted surgery. Springer, pp 283–321. https://doi.org/10.1007/8415_2012_125
30. Rausch MK, Genet M, Humphrey JD (2017) An augmented iterative method for identifying a stress-free reference configuration in image-based biomechanical modeling. J Biomech 58:227–231. https://doi.org/10.1016/j.jbiomech.2017.04.021
31. Mooney M (1940) A theory of large elastic deformation. J Appl Phys 11(582):227–231. https://doi.org/10.1063/1.1712836
32. Rivlin RS (1948) Large elastic deformations of isotropic materials IV. Further developments of the general theory. Philos Trans R Soc Lond Ser A Math Phys Sci 241(835):227–231. https://doi.org/10.1098/rsta.1948.0024
33. Ogden RW (1984) Non-linear elastic deformations. Eng Anal. https://doi.org/10.1016/0264-682X(84)90061-3
34. de Borst R, Crisfield MA, Remmers JJC, Verhoosel CV (2012) Nonlinear finite element analysis of solids and structures, 2nd edn. Willey, New York
35. Krishnamoorthy A, Menon D (2013) Matrix inversion using Cholesky decomposition. In: 2013 signal processing: algorithms, architectures, arrangements, and applications (SPA), pp 70–72
36. Franca LP, Hughes TJR (1988) Two classes of mixed finite element methods. Comput Methods Appl Mech Eng 69(1):89–129. https://doi.org/10.1016/0045-7825(88)90168-5

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.