Real2Sim: Visco-elastic parameter estimation from dynamic motion

DAVID HAHN, ETH Zürich
POL BANZET, ETH Zürich
JAMES M. BERN, ETH Zürich
STELIAN COROS, ETH Zürich

This paper presents a method for optimizing visco-elastic material parameters of a finite element simulation to best approximate the dynamic motion of real-world soft objects. We compute the gradient with respect to the material parameters of a least-squares error objective function using either direct sensitivity analysis or an adjoint state method. We then optimize the material parameters such that the simulated motion matches real-world observations as closely as possible. In this way, we can directly build a useful simulation model that captures the visco-elastic behaviour of the specimen of interest. We demonstrate the effectiveness of our method on various examples such as numerical coarsening, custom-designed objective functions, and of course real-world flexible elastic objects made of foam or 3D printed lattice structures, including a demo application in soft robotics.

CCS Concepts: • **Computing methodologies → Physical simulation**; *Motion processing*.

Additional Key Words and Phrases: Deformable Models, Motion Capture, Optimization, Physically-based Simulation, Robotics

ACM Reference Format:
David Hahn, Pol Banzet, James M. Bern, and Stelian Coros. 2019. Real2Sim: Visco-elastic parameter estimation from dynamic motion. *ACM Trans. Graph.* 38, 6, Article 236 (November 2019), 13 pages. https://doi.org/10.1145/3355089.3356548

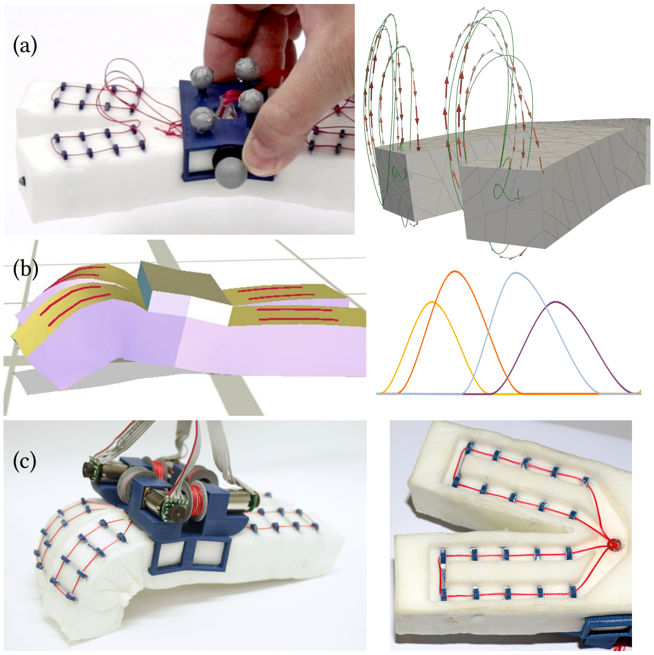

Fig. 1. Application of our method in a soft robotics design pipeline: after fabricating a specimen (a), we capture its behaviour and estimate material parameters; green curve shows motion capture data, arrows show simulation result with optimized parameters. Using these material parameters, (b) we can now design a control strategy in simulation (right image shows motor control curves), and then (c) execute the motion on the real-world robot (right image shows tendons along its underside).

1 INTRODUCTION

Creating fast simulation models that accurately capture the deformation behaviour of real-world objects is not an easy task. Even if physical material parameters are known (for instance from specialized measurement tools) they may not immediately translate to a practical simulation due to various other issues. These problems include limitations of the chosen numerical method, necessity of high-resolution discretizations to achieve accurate solutions, and mismatches between the tested and simulated deformation range for non-linear materials. Many soft materials also exhibit large variations in visco-elastic parameters across multiple objects fabricated from the same base material due to inconsistencies in the fabrication process.

In the face of these challenges, we aim to answer the following question: how can we find a simulation model that (a) captures the deformation behaviour of interest, (b) for a non-linear and possibly non-homogeneous material, (c) at a practical resolution?

The classical approach to address this question is to first measure a small sample of the material using standardized tests, then build a simulation model of sufficient resolution to reproduce the measured data, and finally apply that model to the actual object of interest. Achieving satisfactory results requires high accuracy in each of these steps. Furthermore, many material testing devices, such as rheometers, only measure the linear deformation range, i.e. the material's small-strain behaviour. How to generalize these measurements to the non-linear range is not entirely clear for many materials. Overall, even if an accurate simulation model can be constructed in this way, it might not work for fast, low-resolution simulations due to numerical errors and artefacts, requiring additional corrections such as numerical coarsening before reaching a practically useful simulation.

Following recent works on parameter estimation, we propose a framework to directly fit a general visco-elastic finite element simulation to real-world *motion* data. While various factors, such as resolution, discretization, time-integration method, etc. will influence the resulting fitted material parameters, we aim for the best match to real-world data that a particular choice of simulation method can achieve.

Authors' addresses: David Hahn, CRL, ETH Zürich, david.hahn@inf.ethz.ch; Pol Banzet, CRL, ETH Zürich, pol.banzet@inf.ethz.ch; James M. Bern, CRL, ETH Zürich, james.bern@inf.ethz.ch; Stelian Coros, CRL, ETH Zürich, scoros@inf.ethz.ch.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2019 Copyright held by the owner/author(s).
0730-0301/2019/11-ART236
https://doi.org/10.1145/3355089.3356548

Even if ground-truth material parameters are known, they are useful only when applied to high-resolution models as numerical stiffening can drastically affect simulation results. In our results, we find physically meaningful parameters for standard material models and reasonably small deformations. Furthermore our formulation provides the flexibility to work on custom-designed material models as well. With our approach, we directly obtain a simulation model suitable for a reasonably coarse mesh, the actual specimen, and the deformation and motion range of interest. We envision that this work can facilitate model-based applications in soft robotics control or soft object manipulation; see Fig. 1.

2 RELATED WORK

In this work, we employ a standard finite element method (FEM) to simulate the behaviour of visco-elastic solid objects. In particular, we discretize these objects in material space (i. e. the undeformed configuration) with linear tetrahedral elements. As such formulations are very common in computer graphics, we refer the reader to Irving et al. [2004], Teran et al. [2005], Sifakis and Barbic [2012], and Stomakhin et al. [2012] for further details.

Modelling the stress-strain response of elastic solids has led to a large variety of constitutive equations, such as co-rotational linear elasticity [Felippa and Haugen 2005; Georgii and Westermann 2008; Müller and Gross 2004; Wang et al. 2015], the Saint-Venant-Kirchhoff model [Barbič and James 2005; O'Brien and Hodgins 1999], or the Neohookean material model [Bonet and Wood 2008; Smith et al. 2018], as well as numerous other formulations and variations; see also [Ogden 1997]. Similarly, viscous damping effects due to internal friction have been modelled via either linear (Newtonian) or non-linear constitutive equations relating shear rate to viscous stress [Bower et al. 1993]. One non-linear example is the power-law (or de Waele-Ostwald) model. For traditional material testing methods, the presence of viscous effects may require additional corrections to the measurements as described by [Ngan et al. 2005].

Apart from those well-established material models, recent research in computer graphics has also investigated methods for formulating richer spaces of material responses, both in terms of elasticity [Xu et al. 2015] as well as viscosity [Xu and Barbič 2017]. While we focus mostly on standard material models using physical parameters, we demonstrate that our method can be used in conjunction with such a generalized model in Figure 14.

When simulating a real-world material, not only do we need to formulate a material model, we must also choose its parameters to best approximate experimental observations. As opposed to direct measurements of material parameters, such as the method described by Oliver and Pharr [1992], parameter estimation tries to identify material parameters by matching a simulation model to observed behaviour. The recent review by Mahnken [2017] classifies parameter estimation (or identification) methods into four groups: "hand fitting", "trial-and-error", neural networks, and least-squares fitting. The focus of that work, similar to our approach, is mostly on fitting material parameters by solving a minimization problem on the squared difference between observed and simulated data. Common formulations of the direct sensitivity analysis and adjoint method, as well as standard optimization methods are also summarized there.

Within the computer graphics literature, Pai et al. [1999; 2001] describe a robotic measurement system to collect data from real-world specimens and build a linear elastic model based on boundary Green's functions. In contrast, Bickel et al. [2009] describe how to estimate parameters for elastostatic non-linear materials using a finite element method. They use an optical, marker-based surface reconstruction method to measure the resulting deformation of the object's surface. Both of the aforementioned works rely on experimental measurements of contact forces using custom-designed force sensors, whereas in this paper, we aim to estimate parameters solely based on non-intrusive observations of dynamic motion. Miguel et al. [2012] also use a custom-built deformation capture and force measurement system for cloth samples and then estimate elastostatic parameters in a step-by-step update procedure. They describe the non-linear material response via spline based stress-strain curves and adapt the spline control points automatically. In their follow-up work [Miguel et al. 2013] they extend this approach to include internal friction using force-deformation measurements based on weights attached to the object of interest. They also present an extension to 3D materials instead of cloth [Miguel et al. 2016], again estimating parameters for elastostatic deformations, employing direct sensitivity analysis, while using finite differencing for some terms. Recently, Wang et al. [2015] describe a complete 3D reconstruction and parameter estimation pipeline, where a laser-scanner is used to capture the object's geometry and a multi-camera setup provides dynamic motion data. They then fit a basic co-rotational linear elastic model as well as Rayleigh damping coefficients to match the observed motion in simulation. In doing so, they avoid any gradient computations and instead use a gradient-free Nelder-Mead optimization algorithm. Consequently, their method is not well suited to optimizing inhomogeneous parameter distributions.

Recently, Torres et al. [2016] presented a numerical coarsening scheme for co-rotational linear elasticity including treatment of high-resolution boundary conditions. For non-linear materials, Chen et al. [2015] introduced a data-driven coarse numerical simulation approach for object design and fabrication. They later included dynamic motions as well, introducing dynamics-aware numerical coarsening [Chen et al. 2017]. Their approach matches low-frequency modes of a high-resolution simulation by adjusting the material parameters of a low-resolution mesh in order to preserve the behaviour of corresponding modes up to a specified tolerance. They also use this approach for parameter estimation to match the oscillation frequencies and damping observed in real-world materials via high-speed video capture of optical markers. Even though they use a non-linear material model, they determine material parameters for eigenmode oscillations, which implies a linearisation around a particular configuration (they usually choose the rest shape). In contrast, our approach can deal with arbitrary motions, including forced oscillations. While numerical coarsening is not our primary goal, we show a basic example in Figure 2.

For fluid simulation, McNamara et al. [2004] solve a control problem finding minimal ghost forces such that the simulation produces a desired motion. They employ the adjoint method to derive the gradient of an objective function with respect to these control forces. Their derivation of the adjoint method operates on the time-discretized update equations for advection and pressure projection.

In this way, the time discretization must be chosen before deriving the adjoint operations. Wojtan et al. [2006] employ a similar formulation for keyframe control of particle systems, such as fluids, flocks, and cloth modelled as a mass-spring system.

Bradley [2013], on the other hand, derives the adjoint equations in the continuous setting, leaving the choice of discretization scheme open at this point. We also follow this approach and summarize the resulting equations in Section 3. Auzinger et al. [2018] also successfully employed a similar formulation to optimize 3D printed nanostructures for desired light transport properties such as refraction.

Bern et al. [2019] use a direct sensitivity analysis approach to optimize for cyclic locomotion trajectories of soft robots. We use their simulation framework combined with our parameter estimation system for the demo application shown in Fig. 1.

In summary, we present the following contributions in this paper:

- An optimization-based material parameter estimation method for arbitrary (differentiable) visco-elastic material models,
- using dynamic motion data without the need for special testing equipment or force sensors,
- resulting in efficient simulation models for complex real-world soft objects.

3 THEORY

In this section, we start from a basic finite element method (FEM) for elastodynamic simulation. We then derive the gradient of an objective function that measures the quality of the entire simulation trajectory using first direct sensitivity analysis and then the adjoint state method. As a result we obtain time-continuous, but spatially discretized dynamical systems allowing us to compute the objective function gradient. While these equations can in principle be solved with any suitable time integration method, we show their time discretization using the second order backward differentiation formula (BDF2). We choose BDF2 because it is relatively easy to implement and reduces numerical damping compared to a standard backward Euler (BDF1) scheme, while retaining its A-stability.

3.1 Simulation and optimization

Our starting point is a standard FEM discretization of Newtonian mechanics. Discretizing in space first results in the following system of first-order ordinary differential equations in time, where $\mathbf{x}$ contains node positions of the mesh, $\mathbf{v}$ their velocities, and $\mathbf{f}$ the internal forces acting on them:

$$\begin{aligned} \dot{\mathbf{x}} &:= \mathrm{d}\,\mathbf{x}/\mathrm{d}\,t = \mathbf{v}, \\ \mathbf{M}\dot{\mathbf{v}} &= \mathbf{f}(\mathbf{x},\mathbf{v}) + \mathbf{f}_{\text{ext}}, \end{aligned} \tag{1}$$

with suitable initial and boundary conditions, which we will discuss in more detail later; see also Eq. (13). The external forces, $\mathbf{f}_{\text{ext}}$, are independent of the simulation state; in our examples, we only consider gravity as external load.

The BDF2 integration rule [Iserles 1996] for an ODE of the form $\dot{y} = h(t, y)$ is $y^{i+1} = 4y^i/3 - y^{i-1}/3 + 2\Delta_t h(t^{i+1}, y^{i+1})/3$, where Δ_t is a constant time step. In terms of the coupled dynamics system (1), we choose to first express the end-of-time-step positions as

$$\mathbf{x}^{i+1} = 4\mathbf{x}^i/3 - \mathbf{x}^{i-1}/3 + 2\Delta_t\mathbf{v}^{i+1}/3$$

and then solve

$$\begin{aligned} \frac{3}{2\Delta_t}\mathbf{M}\mathbf{v}^{i+1} = \frac{3}{2\Delta_t}\mathbf{M}\left(\frac{4}{3}\mathbf{v}^i - \frac{1}{3}\mathbf{v}^{i-1}\right) \\ + \mathbf{f}(\mathbf{x}^{i+1}, \mathbf{v}^{i+1}) + \mathbf{f}_{\text{ext}} \end{aligned} \tag{2}$$

for the unknown velocities. The internal forces $\mathbf{f}(\mathbf{x}^{i+1}, \mathbf{v}^{i+1})$ depend non-linearly on the end-of-time-step state and we solve Eq. (2) with a Newton-Raphson iteration. In each iteration the linearized system matrix is found by differentiation with respect to the unknown velocities $\mathbf{v}^{i+1}$, which yields $\mathbf{S} := 3\mathbf{M}/(2\Delta_t) + 2\Delta_t\mathbf{K}/3 + \mathbf{D}$. We define the usual tangent stiffness and damping matrices $\mathbf{K} := -\partial\mathbf{f}/\partial\mathbf{x}$ and $\mathbf{D} := -\partial\mathbf{f}/\partial\mathbf{v}$. Note that $\mathbf{K}$ and $\mathbf{D}$ depend on the current iterate for non-linear elastic or damping models respectively.

We now assume that the internal forces depend on a vector of (material) parameters $\mathbf{q}$, i. e. $\mathbf{f}(\mathbf{x}, \mathbf{v}, \mathbf{q})$ and consequently the simulation state also becomes parameter dependent, i. e. $\mathbf{x}(t, \mathbf{q}), \mathbf{v}(t, \mathbf{q})$. The ODE system (1) can be viewed as a constraint due to physics. Our goal is to optimize an objective function subject to this constraint:

$$\begin{aligned} \mathbf{q}^* &= \arg\min\ \Phi(\mathbf{q}),\ \text{where} \\ \Phi(\mathbf{q}) &:= \int_{t=0}^{T} \varphi(\mathbf{x}, \mathbf{v}, t, \mathbf{q})\ \mathrm{d}\,t, \end{aligned} \tag{3}$$

and $\{\mathbf{x}(t, \mathbf{q}), \mathbf{v}(t, \mathbf{q})\}$ satisfy Eq. (1) for any given, time-independent parameter vector $\mathbf{q}$.

In order to solve this minimization problem, we want to compute the objective function gradient with respect to the parameters $\mathbf{q}$:

$$\frac{\mathrm{d}\,\Phi}{\mathrm{d}\,\mathbf{q}} = \int_{t=0}^{T} (\frac{\partial\varphi}{\partial\mathbf{x}}\frac{\mathrm{d}\,\mathbf{x}}{\mathrm{d}\,\mathbf{q}} + \frac{\partial\varphi}{\partial\mathbf{v}}\frac{\mathrm{d}\,\mathbf{v}}{\mathrm{d}\,\mathbf{q}} + \frac{\partial\varphi}{\partial\mathbf{q}})\ \mathrm{d}\,t. \tag{4}$$

3.2 Direct sensitivity analysis

In the direct sensitivity analysis approach, as the name suggests, we compute the above integral directly. The *sensitivities* $\mathbf{s}_x := \mathrm{d}\,\mathbf{x}/\mathrm{d}\,\mathbf{q}$, and $\mathbf{s}_v := \mathrm{d}\,\mathbf{v}/\mathrm{d}\,\mathbf{q}$ must satisfy an ODE that follows from differentiating Eq. (1) with respect to $\mathbf{q}$, which yields

$$\begin{aligned} \dot{\mathbf{s}}_x &= \mathbf{s}_v, \\ \mathbf{M}\dot{\mathbf{s}}_v &= -\mathbf{K}\mathbf{s}_x - \mathbf{D}\mathbf{s}_v + \partial\mathbf{f}/\partial\mathbf{q}. \end{aligned} \tag{5}$$

Here we assume that the mass matrix and external forces are independent of the parameters for brevity. We can remove this assumption by including the additional derivative terms outlined in Section 4. This system of ODEs for the sensitivities can be integrated forward in time along with the simulation of (1). The matrices $\mathbf{K}$ and $\mathbf{D}$ are derivatives with respect to the simulation state, and therefore assume that the physics constraint is fulfilled and the corresponding state is known. Consequently, we can integrate the sensitivities after computing each time step of the forward simulation, using the stiffness and damping matrices corresponding to the state computed for the end of that time step. As a result, even though the internal forces may depend non-linearly on the state, Eq. (5) remains a system of *linear* ODEs in $\mathbf{s}_x$ and $\mathbf{s}_v$, and the sensitivities are always found in a single linear solve per time step in practice.

The time discretization for the direct sensitivity approach follows in the same way as Eq. (2), except that Eq. (5) is already linear and the simulation states, as well as the stiffness and damping matrices, are already known. Consequently, the BDF2 discretization results in the following linear system, using the same system matrix as the forward simulation (evaluated for the end-of-time-step state):

$$\begin{aligned} \mathbf{s}_x^{i+1} &= 4\mathbf{s}_x^{i}/3 - \mathbf{s}_x^{i-1}/3 + 2\Delta_t \mathbf{s}_v^{i+1}/3, \\ \mathbf{S}\mathbf{s}_v^{i+1} &= \frac{3}{2\Delta_t}\mathbf{M}\left(\frac{4}{3}\mathbf{s}_v^{i} - \frac{1}{3}\mathbf{s}_v^{i-1}\right) \\ &\quad - \mathbf{K}\left(\frac{4}{3}\mathbf{s}_x^{i} - \frac{1}{3}\mathbf{s}_x^{i-1}\right) + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}. \end{aligned} \tag{6}$$

3.3 Adjoint method

The main drawback of the direct method is that the computational cost of finding the objective function gradient in (4) increases with the number of parameters. The adjoint method provides an elegant way of reducing this computational cost; see also [McNamara et al. 2004] and [Bradley 2013]. In order to derive the adjoint equations, we first write the state of the simulation as a concatenated vector $\mathbf{y} := (\mathbf{x}^\mathsf{T}\ \mathbf{v}^\mathsf{T})^\mathsf{T}$. Then we rewrite Eq. (1) as $\mathbf{h}(\mathbf{y}, \dot{\mathbf{y}}, t, \mathbf{q}) = \mathbf{0}$, where

$$\begin{aligned} \mathbf{h} :=& \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{y} \\ &+ \begin{pmatrix} \mathbf{0} \\ \mathbf{f}(\mathbf{y}) + \mathbf{f}_\text{ext} \end{pmatrix} - \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{pmatrix} \dot{\mathbf{y}}. \end{aligned} \tag{7}$$

Bradley [2013] describes how to derive the general form of the adjoint method for this case, which reads

$$\frac{\partial \varphi}{\partial \mathbf{y}}^\mathsf{T} + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{y}}^\mathsf{T} - \frac{\mathrm{d}}{\mathrm{d}\,t}\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{y}}}^\mathsf{T}\right)\boldsymbol{\lambda} - \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{y}}}^\mathsf{T} \dot{\boldsymbol{\lambda}} = 0, \tag{8}$$

where $\boldsymbol{\lambda}$ is referred to as the adjoint state. The objective function gradient then becomes

$$\frac{\mathrm{d}\,\Phi}{\mathrm{d}\,\mathbf{q}} = \int_{t=0}^{T} \left(\frac{\partial \varphi}{\partial \mathbf{q}} + \boldsymbol{\lambda}^\mathsf{T} \frac{\partial \mathbf{h}}{\partial \mathbf{q}}\right) \mathrm{d}\,t, \tag{9}$$

plus a term that depends on the initial conditions of (1); see Eq. (14).

The derivatives required to evaluate Eq. (8) follow from Eq. (7):

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{y}}}^\mathsf{T} &= -\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^\mathsf{T} \end{pmatrix}, \\ \frac{\mathrm{d}}{\mathrm{d}\,t}\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{y}}}^\mathsf{T} &= \mathbf{0}, \text{ and} \\ \frac{\partial \mathbf{h}}{\partial \mathbf{y}}^\mathsf{T} &= \begin{pmatrix} \mathbf{0} & -\mathbf{K}^\mathsf{T} \\ \mathbf{I} & -\mathbf{D}^\mathsf{T} \end{pmatrix}. \end{aligned} \tag{10}$$

Note that in the last term, we need to take derivatives of the internal forces with respect to the entire state, which means first w. r. t. positions and then w. r. t. velocities, resulting again in the tangent stiffness and damping matrix respectively. Furthermore, in a Galerkin FEM discretization the mass, stiffness, and damping matrices are symmetric. Nevertheless, we maintain transpositions for the sake of clarity.

Finally, separating the adjoint state into two parts, analogous to the definition of the concatenated state $\mathbf{y}$, corresponding to adjoint states for positions and velocity respectively $\boldsymbol{\lambda} =: (\boldsymbol{\lambda}_x^\mathsf{T}\ \boldsymbol{\lambda}_v^\mathsf{T})^\mathsf{T}$ and splitting (8) into two equations accordingly produces

$$\begin{aligned} -\dot{\boldsymbol{\lambda}}_x &= (\partial \varphi / \partial \mathbf{x})^\mathsf{T} - \mathbf{K}^\mathsf{T} \boldsymbol{\lambda}_v, \text{ and} \\ -\mathbf{M}^\mathsf{T} \dot{\boldsymbol{\lambda}}_v &= (\partial \varphi / \partial \mathbf{v})^\mathsf{T} + \boldsymbol{\lambda}_x - \mathbf{D}^\mathsf{T} \boldsymbol{\lambda}_v. \end{aligned} \tag{11}$$

Equation (11) is again a system of linear first-order ODEs for the adjoint states. However, the "initial" conditions differ in one important aspect: while we assume the usual initial conditions for the forward simulation (1) of the form $\mathbf{x}(t = 0) = \mathbf{x}_0$ and $\mathbf{v}(t = 0) = \mathbf{v}_0$, the derivation of the adjoint formulation requires that $\boldsymbol{\lambda}_x(t = T) = \boldsymbol{\lambda}_v(t = T) = 0$ at the final time T of the simulation; see [Bradley 2013] for details. Consequently, the ODE system in Eq. (11) must be integrated *backwards* in time after the forward simulation has been completed.

Applying the BDF2 integration rule backwards in time means the sign of the time-derivative term changes, which cancels the minus signs on the left-hand sides of (11), while Δ_t remains a positive constant. Also note that $\boldsymbol{\lambda}_x^{i-1}$ and $\boldsymbol{\lambda}_v^{i-1}$ are now the *unknown* adjoint states, so time indices $(i - 1)$ and $(i + 1)$ must be swapped.

Discretizing (11) in this way yields

$$\begin{aligned} \boldsymbol{\lambda}_x^{i-1} =& \frac{4}{3}\boldsymbol{\lambda}_x^{i} - \frac{1}{3}\boldsymbol{\lambda}_x^{i+1} + \frac{2}{3}\Delta_t\left(\frac{\partial \varphi}{\partial \mathbf{x}}^\mathsf{T} - \mathbf{K}^\mathsf{T}\boldsymbol{\lambda}_v^{i-1}\right), \\ \mathbf{S}^\mathsf{T}\boldsymbol{\lambda}_v^{i-1} =& \frac{3}{2\Delta_t}\mathbf{M}^\mathsf{T}\left(\frac{4}{3}\boldsymbol{\lambda}_v^{i} - \frac{1}{3}\boldsymbol{\lambda}_v^{i+1}\right) + \frac{\partial \varphi}{\partial \mathbf{v}}^\mathsf{T} \\ &+ \frac{4}{3}\boldsymbol{\lambda}_x^{i} - \frac{1}{3}\boldsymbol{\lambda}_x^{i+1} + \frac{2}{3}\Delta_t \frac{\partial \varphi}{\partial \mathbf{x}}^\mathsf{T}. \end{aligned} \tag{12}$$

All data that is required by the adjoint simulation must be stored while performing the forward simulation, leading to a higher memory cost than the direct method described above. The main difference is that the direct method needs to compute and store the sensitivities $\mathbf{s}_x$ and $\mathbf{s}_v$, which are dense $|\mathbf{x}| \times |\mathbf{q}|$ matrices, where $|\mathbf{x}|$ is the number of degrees of freedom in the simulation and $|\mathbf{q}|$ is the number of parameters. While these matrices need to be updated in every time step, the memory cost remains constant over time. The adjoint method on the other hand reduces the computation to the adjoint states, which are only $|\mathbf{x}| \times 1$ vectors. The memory consumption, however, increases linearly over time as we must store additional data in each time step. Consequently, the direct method is preferable for simulations that have very few parameters, but cover a longer time span, while the adjoint method works best for reasonably short simulations, but can deal with a large number of parameters.

3.4 Initial conditions

So far we have assumed that the initial conditions of the forward problem are independent of the parameters $\mathbf{q}$. If this is not the case and the initial conditions can be written in the general form

$$\mathbf{g}(\mathbf{x}, \mathbf{v}, \mathbf{q}, t = 0) = 0, \tag{13}$$

we must account for the additional influence of the parameters on the objective function value that occurs through a change of these initial conditions. One important instance of this situation is starting a dynamic simulation from a state of static equilibrium, that is $\mathbf{f}(\mathbf{x}(t = 0), \mathbf{0}) + \mathbf{f}_\text{ext} = 0$ and $\mathbf{v}(t = 0) = \mathbf{0}$, where the internal forces $\mathbf{f}$ depend on elastic material parameters.

For the direct sensitivity analysis, Eq. (5), the initial conditions on the sensitivities are the derivative of the initial conditions on (1) with respect to the parameters. For the general implicit form, we obtain

$$\frac{\mathrm{d}\,\mathbf{g}}{\mathrm{d}\,\mathbf{q}} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{s}_{x0} + \frac{\partial \mathbf{g}}{\partial \mathbf{v}}\mathbf{s}_{v0} + \frac{\partial \mathbf{g}}{\partial \mathbf{q}} = 0,$$

where subscript zero indicates quantities at $t = 0$, in particular $\mathbf{s}_{x0} := \mathbf{s}_x(t = 0)$ and similarly for $\mathbf{s}_{v0}$. For the elastostatic case this condition simplifies to $\mathbf{s}_{v0} = 0$ and $\mathbf{K}_0\mathbf{s}_{x0} = \partial\mathbf{f}_0/\partial\mathbf{q}$. Consequently, if we find the initial positions by solving for elastostatic equilibrium, we analogously find the initial sensitivities by solving the linearized elastostatic system for the derivative of the internal forces w. r. t. the parameters. Note that $\mathbf{K}_0$ must be evaluated at the configuration $\mathbf{x}_0$ that satisfies the initial conditions, i. e. after solving the initial elastostatic problem.

For the adjoint method, the objective function gradient according to [Bradley 2013] becomes

$$\frac{\mathrm{d}\,\Phi}{\mathrm{d}\,\mathbf{q}} = \mathcal{I}_T + \boldsymbol{\lambda}_0^\mathsf{T} \left(\left.\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{y}}}\right|_{t=0} \right) \left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}_0} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{q}}, \tag{14}$$

where $\mathcal{I}_T$ represents the time integral term in Eq. (9). This additional term simplifies to $(\mathbf{K}_0{}^{-\mathsf{T}}\boldsymbol{\lambda}_{x0})^\mathsf{T}(\partial\mathbf{f}_0/\partial\mathbf{q})$ for *elastostatic* initial conditions. Consequently, similar to the direct method above, we again need to solve the linearized elastostatics problem, this time with the adjoint state as right-hand-side vector.

Finally, note that BDF2 requires initialization of previous values at $t = -\Delta_t$. For the forward simulation and the direct sensitivity analysis we can make the assumption that both velocity and acceleration are zero before $t = 0$. In particular, when starting from an elastostatic equilibrium configuration, it is natural to interpret the initial condition $v(t = 0) = 0$ as $v(t \leq 0) = 0$. Similarly, we need to make an assumption on $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_v$ for $t \geq T$ for the adjoint states. From Eq. (11) we can see that the adjoint states basically encode objective function gradients. Consequently, we again assume zero change outside of the simulated time interval, extending the condition on the adjoint states $\boldsymbol{\lambda}_x(t = T) = \boldsymbol{\lambda}_v(t = T) = 0$ to $\boldsymbol{\lambda}_x(t \geq T) = \boldsymbol{\lambda}_v(t \geq T) = 0$.

4 METHODS AND DETAILS

4.1 Experimental methods

To collect data that captures the dynamical behaviour of our test specimens, we use a commercial optical motion capture system, consisting of 10 *OptiTrack Prime 13* cameras. We use small hemispherical reflective markers of 4 mm diameter to track the soft specimens' deformation. We 3D print rigid clamps that also have motion capture markers attached and glue our specimens to these clamps. In this way we can track the motion of the clamps and map it to Dirichlet boundary conditions in the simulation. We usually use groups of 4 to 6 spherical reflective markers of 1 or 1.4 cm diameter to track the rigid transformation of these boundary clamps. This approach allows us to freely move the clamped boundary during capture and reproduce its motion in the simulation later. Consequently, the simulation and the motion capture data share a common coordinate system. Similar to the boundary conditions, we use the tracked trajectories of the motion capture markers $\mathbf{x}_i^*$ and formulate our objective function based on the distance to "virtual markers" $\bar{\mathbf{x}}_i$ in the simulation:

$$\Phi := \frac{1}{2} \int_{t=0}^{T} \sum\nolimits_i \left\| \mathbf{x}_i^*(t) - \bar{\mathbf{x}}_i(t) \right\|^2 \mathrm{d}\,t, \tag{15}$$

where T is the duration for which we recorded motion data (which in turn determines the simulated time as well) and i is the marker index. The root-mean-square error per marker is then $\sqrt{2\Phi/(NT)}$, where N is the number of tracked markers. While we choose a marker-based approach due to its ease of use and availability, the theory presented so far would equally apply to motion data reconstructed from any other capture system, as long as an objective function on the simulation state can be written in terms of the collected data.

One interesting material with recent applications in soft robotics [Somm et al. 2019] is flexible polyurethane foam. In general this material is relatively cheap and easy to work with. In particular, we use *FlexFoam-iT* III and V from *Smooth-On, Inc.* [2018]. These foams are shipped as two separate liquid components that have to be mixed before moulding; the foam then expands in the mould in about 15 min and cures in a few hours at room temperature. The resulting material parameters of the cured foam can vary drastically between multiple samples of the same base material depending on ambient temperature, precise mixing ratio, as well as the exact amount of material filled into the mould and the resulting pressure build-up during expansion. We produce several test specimens and the soft robot in Fig. 1 from these foams. It is therefore imperative to enable fast parameter estimation for a particular specimen.

We also design a sparse lattice structure using *Autodesk NetFabb* and 3D print it with a flexible TPU material on a *Stratasys* F370 printer [Stratasys, Ltd. 2018]; see Fig. 5a. This particular lattice has a unit cell size of 1 cm and a beam thickness of 1.6 mm.

4.2 Material models

We implement a couple of different material models to test our simulation and optimization framework. While these models are based on previous work, in some cases variations exist in the literature, so we briefly summarize their definitions here. In general, we compute all derivatives using symbolic differentiation unless stated otherwise.

We primarily employ a Neohookean elasticity model. This model is defined via the elastic strain energy density

$$\begin{aligned} \Psi(\mathbf{F}) :=& \frac{\lambda}{2} \left((J^2 - 1)/2 - \ln J \right) \\ &+ \frac{\mu}{2} \left(\mathrm{tr}(\mathbf{F}^\mathsf{T}\mathbf{F}) - 3 - 2\ln J \right). \end{aligned} \tag{16}$$

Here λ is the first Lamé parameter, μ is the shear modulus, $\mathbf{F}$ is the deformation gradient (which is piecewise-constant for a linear tetrahedral FEM), and $J := \det \mathbf{F}$. The nodal force contribution per element is then $\mathbf{f}_{\mathrm{nh}} = -V_e(\mathrm{d}\,\Psi/\mathrm{d}\,\mathbf{x}) = -V_e(\partial\Psi/\partial\mathbf{F})(\mathrm{d}\,\mathbf{F}/\mathrm{d}\,\mathbf{x})$, where V_e is the element's volume. The first Piola-Kirchhoff stress $\partial\Psi/\partial\mathbf{F}$ is found by directly differentiating the energy density function, and the derivative of $\mathbf{F}$ is given by the FEM shape function gradients.

In order to model viscous damping forces, i. e. history-independent damping due to velocity, we implement a power-law viscosity model, where the viscous stress is defined as

$$\boldsymbol{\sigma}_{\mathrm{pl}}(\mathbf{v}) := \nu\dot{\gamma}^{(h-1)}(\nabla\mathbf{v} + \nabla\mathbf{v}^\mathsf{T}), \tag{17}$$

where $\dot{\gamma}$ is the shear rate $\dot{\gamma} := \sqrt{1/2}\left\|\nabla\mathbf{v} + \nabla\mathbf{v}^\mathsf{T}\right\|_F$, ν is the viscosity coefficient, and h is the power-law index. For $h = 1$, this model simplifies to Newtonian viscosity, $h < 1$ describes a shear thinning behaviour, whereas $h > 1$ corresponds to a shear thickening material. In the shear-thinning case, the viscous forces are technically

Table 1. Material parameters estimated for coarse meshes of the Stanford bunny given motion data from a high-resolution ground-truth simulation. Errors are given for the simulations using first the original and then the optimized parameters. Timings were measured on a 3.9 GHz quad-core desktop CPU; the first value gives the runtime of a single simulation (including objective function gradient evaluation), the second value gives the runtime of the entire optimization.

	Ground truth (50k)	8.5k mesh		2.7k mesh		1.2k mesh	
Lamé λ [Pa]	1.20E+04	5.41E+03	45%	5.92E+03	49%	6.30E+03	53%
Lamé μ [Pa]	1.10E+04	9.97E+03	91%	8.74E+03	79%	6.11E+03	56%
Viscosity coefficient [Pa s]	8	8.12	102%	6.79	85%	2.33	29%
Power-law index	0.8	0.98	123%	1.14	143%	1.76	220%
RMS error [mm] (ori/opt)		14.40	3.82	17.16	4.83	15.94	4.85
Runtime [min] (sgl/opt)	14.0	2.5	69.8	0.7	42.0	0.4	25.5

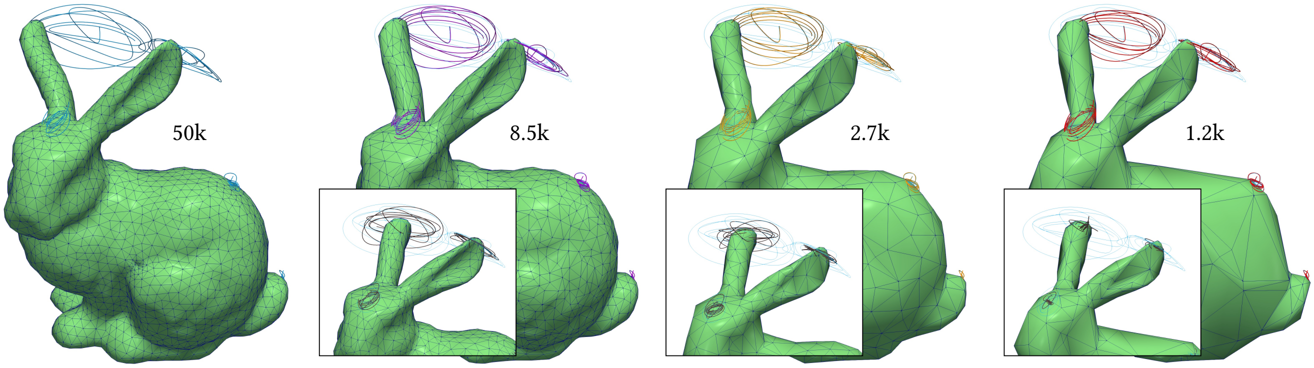

Fig. 2. Simulations of the Stanford Bunny on meshes of various resolutions (labelled by number of elements per mesh). Light blue curves show the ground truth data obtained from the highest resolution mesh; purple, orange, and red curves show the trajectories of virtual markers on both ears, the head, the back, and the tail when using optimized material parameters for each mesh. Black curves in inset images show trajectories obtained with ground-truth parameters on coarser meshes. The motion is driven by a scripted Dirichlet boundary condition on the base; please also refer to our accompanying video.

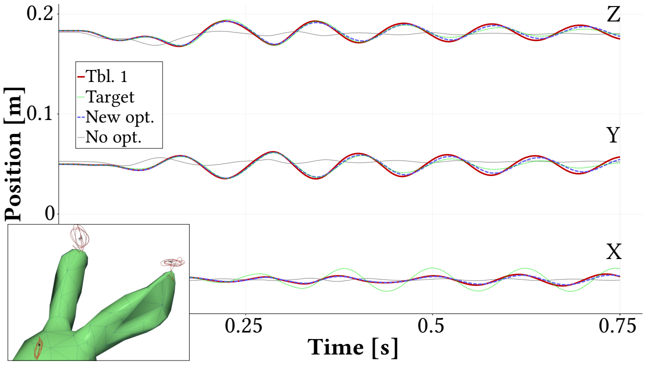

Fig. 3. Evaluation of the optimized parameters on the 1.2k element mesh: we simulate a different motion (red solid lines) using the parameters optimized for the original motion (Table 1). Blue dashes show the result obtained by optimizing for the new motion instead, green dots show the high-resolution ground-truth result for this new motion. Thin black lines show motion graphs when simulating the 1.2k mesh with ground-truth parameters. All curves refer to the bunny's right ear. The inset image shows results for the ground-truth (black) and Table 1 parameters (red) on the coarse mesh.

undefined for a vanishing velocity gradient, but are zero in the limit. In our implementation we simply skip any element where the Frobenius norm of the velocity gradient is less or equal DBL_MIN.

Figures 14 and 15 show results using a principal-stretch based material model as described in [Xu et al. 2015]. We briefly summarize the main features of this material model here. They start by defining the elastic energy density Ψ as a function of principal stretches ς_i, which are computed using a singular value decomposition of the deformation gradient: $\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^\mathsf{T}$, $\operatorname{diag}(\hat{\mathbf{F}}) = (\varsigma_1, \varsigma_2, \varsigma_3)$. The energy density is then given by Eq. (11) in [Xu et al. 2015] as

$$\begin{aligned}\Psi = f(\varsigma_1) + f(\varsigma_2) + f(\varsigma_3) + h(\varsigma_1\varsigma_2\varsigma_3) +\\ g(\varsigma_1\varsigma_2) + g(\varsigma_2\varsigma_3) + g(\varsigma_3\varsigma_1).\end{aligned} \tag{18}$$

They show that a Neohookean model can be re-written in this form by setting

$$\begin{aligned} f(x) = \mu(x^2 - 1)/2, \quad g \equiv 0,\\ h(x) = -\mu \log x + \lambda(\log x)^2/2,\end{aligned} \tag{19}$$

where (λ, μ) are the Lamé parameters as before. Consequently, we always assume $g \equiv 0$ from now on, as well as in our implementation. Finally, the components of the first Piola-Kirchhoff stress $\mathrm{d}\,\Psi/\mathrm{d}\,\mathbf{F}$ can be computed as $(\partial\Psi/\partial\varsigma_i)(\mathrm{d}\,\varsigma_i/\mathrm{d}\,\mathbf{F})$, where

$$\partial\Psi/\partial\varsigma_1 = f'(\varsigma_1) + h'(\varsigma_1\varsigma_2\varsigma_3)\varsigma_2\varsigma_3 \tag{20}$$

and analogously for the other two principal stretches.

Xu et al. [2015] also describe how to compute the required derivatives of the singular value decomposition analytically, as well as how to assemble the corresponding stiffness matrix. Recently, Smith

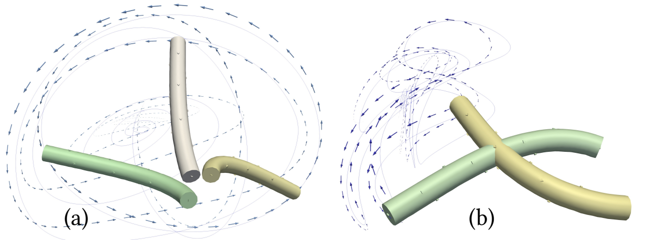

Fig. 4. Motion capture data (solid lines) and simulated data (arrows) for different dynamic motion samples. The coloured cylinders show snapshots of a simulation using optimized material parameters for a Neohookean elastic and power-law viscous material.

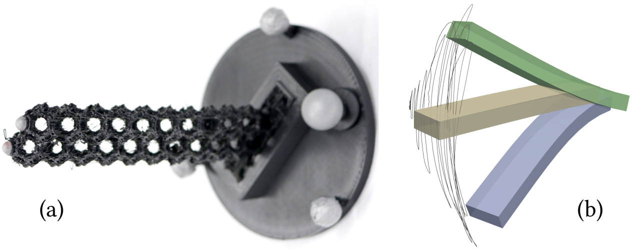

Fig. 5. A 3D printed soft lattice (a) with boundary clamp and motion capture markers and (b) real-world motion trajectory (solid line) with simulation snapshots at different times during the simulation (green: 0.17 s, blue: 0.75 s, beige: 1.4 s). While the real-world specimen is a very complex structure, the simulation uses less than 120 tetrahedral elements.

et al. [2019] also investigated similar analytical derivatives. In our implementation we represent the two stress-strain response functions f' and h' as cubic Hermite splines following the formulation of [Kochanek and Bartels 1984] and we use the *autodiff* module of Eigen [Guennebaud et al. 2010] to compute the derivatives of the spline coefficients with respect to the input control point values. These derivatives are required to describe the change of internal forces due to a change in parameter (i. e. control point) values. For convenience, we also use *autodiff* to compute the SVD derivatives instead of implementing the analytical formulation of [Xu et al. 2015].

4.3 Computational methods

During forward simulation, we solve Eq. (2) with a standard Newton-Raphson method; we iterate until the residual force magnitude is less than 10^{-6} N, or at most 20 iterations (most examples converge within 3 to 6 iterations per time step). We use a stabilized bi-conjugate gradient solver for the sparse linear sub-problem within each iteration; we find this solver to produce numerically more robust results compared to a standard conjugate gradient version. Our implementation builds on the linear algebra library Eigen [Guennebaud et al. 2010]; we use their default solver configuration for all of our results. We store the solver object, which encapsulates the system matrix, at the end of each time-step and re-use it to solve the linear problem for either the sensitivities in Eq. (6) or the adjoint states in Eq. (12).

We use the LBFGS algorithm [Nocedal 1980] to solve the parameter optimization problem, Eq. (3); our implementation is based on the publicly available LBFGS++ library [Qiu 2019]. The optimization algorithm is limited to at most 150 iterations, with at most 20 function evaluations (i. e. simulation runs) per iteration. In each iteration we perform a line search using regular Wolfe conditions with curvature coefficient 0.999.

The physical material parameters we optimize for are usually expected to be non-negative; furthermore their values can have dramatically different orders of magnitude. For instance, the power-law index h is expected to be close to 1, whereas the shear modulus could easily be on the order of 10^6 Pa. We found that in these situations optimizing for the logarithm of the parameters, instead of the parameter values directly, can greatly improve the convergence of the optimization algorithm and ensures positivity without additional constraints. Figure 11 compares the convergence behaviour of a homogeneous material parameter estimation with and without this logarithm transform on a very simple example. When optimizing for the log of parameters, the objective function decreases faster and further compared to optimizing for the parameter values directly. For more complex, real-world motions, attempting to optimize for the parameter values directly can sometimes fail completely. Obviously, transforming the parameters also results in a transformation of derivatives as follows: if we replace parameter q_i with $\tilde{q}_i := \ln q_i$, then $q_i = \exp(\tilde{q}_i) = \mathrm{d}\,q_i/\mathrm{d}\,\tilde{q}_i$ and consequently for any quantity y depending on this parameter we find $\mathrm{d}\,y/\mathrm{d}\,\tilde{q}_i = (\mathrm{d}\,y/\mathrm{d}\,q_i)\ q_i$.

We can also treat density as an input parameter (either globally homogeneous, or locally per element). As the density influences both the mass matrix and the external force due to gravity, we need to account for the additional terms $\mathrm{d}\,\mathbf{f}_{\text{ext}}/\mathrm{d}\,\mathbf{q}$ and $(\mathrm{d}\,\mathbf{M}/\mathrm{d}\,\mathbf{q})\dot{\mathbf{v}}$ and add them in Eq. (5) and (9). We have skipped these terms in Section 3 for the sake of brevity; extending the derivations to include them is straightforward.

5 RESULTS

In this section we present results obtained with our method. We start with some basic verification tests using a ground-truth simulation as the target motion (Figure 2). Our goal is to find parameters for standard material models from dynamic motions. We extensively test our method on cylindrical specimens using various motion samples, including oscillations in the first and second harmonic modes (Fig. 4), as well as strong bending deformation (Fig. 8). Similarly, we also match the fast oscillation of a flexible 3D printed lattice structure simulated with a very coarse mesh (Fig. 5). Finally, the adjoint method allows us to optimize local parameters per element. We test our approach on a manually designed deformation target (Fig. 10), as well as on an inhomogeneous real-world specimen (Fig. 13). For all of these examples, we usually optimize for the logarithm of the material parameters as described in Section 4. Figure 11 shows a comparison of convergence behaviour with and without applying the log-transform on a simple test case.

As an introductory example, we first show how our method can be used to adjust material parameters for a sequence of coarser meshes, counteracting increasing numerical stiffness and damping as the resolution reduces. We first run a ground-truth, high-resolution simulation (ca. 50k tetrahedra) on the Stanford bunny with a homogeneous Neohookean material and power-law viscous damping. In this example the bunny is actuated by a moving Dirichlet boundary condition on its base, performing a scripted motion composed of

Table 2. Material parameters estimated from motion capture data for single-type foam cylinder specimens actuated by one boundary clamp. The simulation uses a Neohookean elastic and power-law viscous material. Timings show the CPU time (4 × 3.9 GHz) for a single simulation with objective function gradient evaluation first, then the runtime for the entire optimization.

Foam type	III	III	III	V	V	V	V
Figure				4a	4b		8
Lamé λ [Pa]	1.17E+04	1.17E+04	1.18E+04	1.28E+04	1.23E+04	1.25E+04	1.67E+04
Lamé μ [Pa]	1.27E+05	1.26E+05	1.10E+05	1.50E+05	1.53E+05	1.57E+05	9.80E+05
Viscosity coefficient [Pa s]	8.74	8.64	7.28	1.94	8.51	3.65	0.20
Power-law index	0.44	0.57	0.49	0.98	0.49	0.78	2.95
Runtime [min] (sgl/opt)	1.2 / 76.5	3.8 / 263.9	1.4 / 44.9	1.5 / 111.5	1.7 / 181.1	3.5 / 209.3	1.1 / 46.6

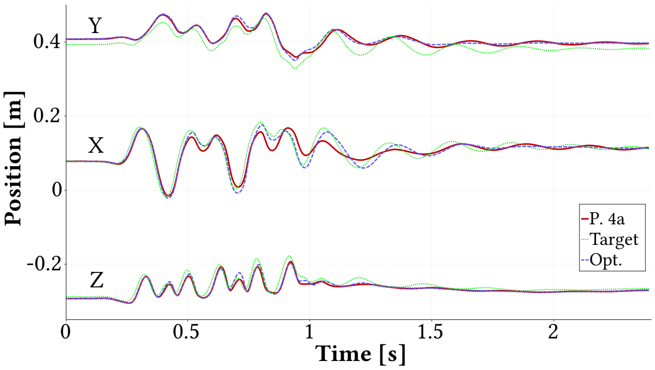

Fig. 6. Comparison of parameters: motion graphs for the example of Fig. 4b simulated with parameters optimized for the motion of Fig. 4a (red solid lines). Green dots show the recorded target motion of the physical specimen; blue dashes show the optimal result for this motion. All curves refer to the top marker on the unconstrained end of the specimen.

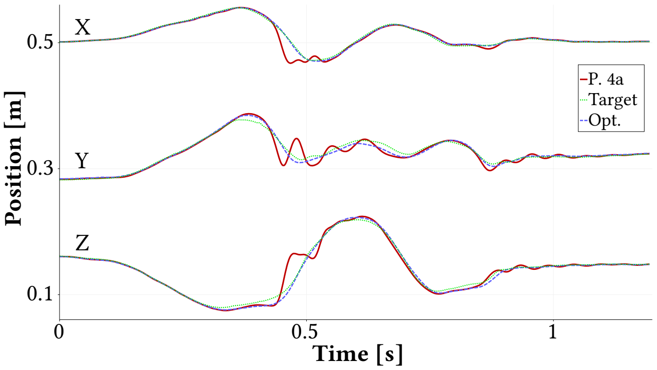

Fig. 7. Comparison of parameters: motion graphs for the example of Fig. 8 simulated with parameters optimized for the motion of Fig. 4a (red solid lines). Green dots show the recorded target motion of the physical specimen; blue dashes show the optimal result for this motion. All curves refer to the marker in the middle of the specimen as in Fig. 8.

two sinusoid displacements oscillating at 17Hz and 13Hz respectively. We record the motion trajectories of five "virtual" markers due to this actuation as shown in Fig. 2. Our method then finds optimal material parameters for three coarser meshes (8.5k, 2.7k, and 1.2k elements respectively) such that the trajectories of these virtual markers match the ground-truth simulation as closely as possible. Each optimization runs independently using the ground-truth data as initial parameter values. As coarser elements introduce numerical stiffness, the optimization in turn results in softer material parameters than the ground truth, as expected.

The resulting material parameters are summarized in Table 1. The bulk modulus drops to about 62.4%, 60.7%, and 53.7% of the original value respectively, while the shear modulus drops to 90.6%, 79.4%, and 55.5% respectively. The damping parameters move from the initial shear-thinning behaviour towards shear-thickening parameters while the viscosity coefficients drop to just under 30% on the coarsest mesh as numerical damping increases. As shown in Fig. 2, the motion trajectories of the ears are noticeably different between the three coarse versions and the high-resolution version, whereas the trajectories for the head, back, and tail match up quite well. We suspect that the noticeable differences between the ear trajectories are due to heavy numerical stiffening around the part that has the smallest cross-section area just above the head. While one might expect lowering the stiffness further would help to get the trajectories closer to the target motion, at some point the resonant behaviour will be lost, so the optimal result shows a compromise between these cases.

We then test whether the material parameters optimized for the original motion (Fig. 2) generalize to a different one using the coarsest mesh in both cases. In the original case, the boundary condition on the base of the bunny moves in the x-y-plane, whereas the new motion proceeds in the y-z-plane with similar oscillation frequencies. Using the ground truth material parameters for the coarse mesh leads to an RMS error of 6.7 mm, whereas the optimal result for this motion yields 2.6 mm. Using the parameters optimized for the original motion results in an RMS error of 2.9 mm, and a very similar motion as the optimal parameters as shown in Fig. 3. The optimal shear modulus differs by about 5.5% between the two motions.

Our primary goal is to estimate parameters of real-world materials. We first focus on (mostly) homogeneous specimens, where we only need to optimize for a small number of parameters. In particular, we aim to find the stiffness in terms of the Lamé parameters of a Neohookean elastic model, as well as the parameters of a power-law damping model.

We fabricate two cylindrical specimens, 20 cm tall and 2 cm in diameter, made of *FlexFoam* type III and type V respectively. We place hemispherical motion capture markers of 4 mm diameter on the surface of the cylinders. These markers are represented by slightly

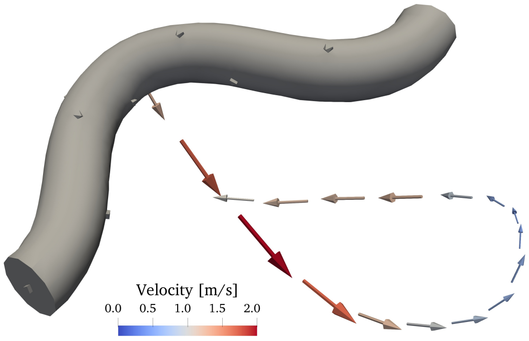

Fig. 8. An example showing large bending deformation and fast “snap-through” motion using two boundary clamps. For this situation the material parameters of a Neohookean model need to be much stiffer compared to the slower, less deformed motions in Fig.4.

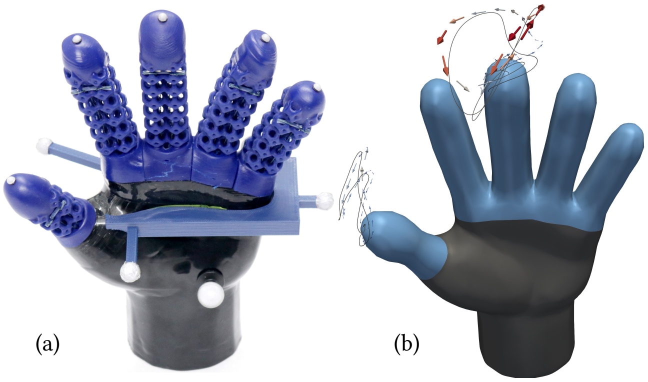

Fig. 9. A 3D printed soft hand (a) with boundary clamp and motion capture markers and (b) simulated (arrows) and real-world motion trajectories (solid) of the thumb and middle finger. The objective function measures trajectories of all five fingers.

elevated triangles on the surface of the simulation mesh (see also Fig. 8). For the cylindrical specimens we use data from at most 9 markers, spaced evenly along the surface, as well as one on the unconstrained end of the cylinder. For the soft robot in Fig. 1 we use only 4 markers, one on the end of each leg. We have not experimented with more complex marker placement patterns as we position them by hand in our experiments. In order to determine the optimal material parameters, we minimize the squared distance between the simulation and real-world marker positions, as observed by the motion capture system; see Eq. (15).

Figure 4 shows example simulations obtained for the type V foam specimen. Note that we find very similar material parameters independently for both motion samples: the first motion (Fig. 4a) oscillates in the object’s base harmonic mode, whereas the second motion (Fig. 4b) oscillates in the second harmonic mode. The root mean square error per marker is 17.9 mm in example (a) and 13.1 mm in (b); note that we do not tune possible constant offsets of marker positions in any way. Further results are summarized in Table 2. The initial parameter values for all these optimizations are $\lambda = 10\,\text{kPa}$, $\mu = 20\,\text{kPa}$, $\nu = 12$ Pa s, and $h = 1$, which means the material is initially simulated softer and overdamped compared to the optimal results. For these homogeneous cases we have not observed notable differences in the optimal results due to different initial values.

We also tested two separate small material samples of the type V foam in a *Paar Physica MCR300* rheometer to measure their shear modulus under small-strain forced oscillations. At 15 Hz actuation frequency the measured shear modulus was 175 kPa and 186 kPa for the two samples respectively. In general, a higher actuation frequency resulted in increased shear moduli, whereas lower frequencies were unreliable due to limitations of the testing equipment. In contrast the motion shown in Fig. 4a corresponds to roughly 5 Hz and Fig. 4b to about 10 Hz. Nevertheless, our results of around 150 kPa are acceptably close to these measurements, especially considering the variations between different specimens made of the same foam type. In Fig. 6 we compare the simulated motion of Fig. 4b using either the parameters optimized for the motion of Fig. 4a or 4b respectively. Even though the optimal damping parameters for these two motions seem quite different, damping is fairly low overall, whereas the stiffness is quite similar for both cases and the motion is simulated well with both parameter sets.

Fitting a two-parameter damping model to these motions containing a single predominant frequency seems to be somewhat ill-conditioned, in the sense that lowering the power-law index while increasing the viscosity may yield very similar results. Our third motion sample for the type-V foam specimen (see video) combines oscillations at two different frequencies and yields damping parameters in a similar range, see also Table 2. We show further comparisons in our accompanying video, including motions using (a) the initial guess parameters that are generally too soft and overdamped, (b) using parameters optimized for another motion sample as in Fig. 6, as well as (c) an example with stiffness increased by 20% over the optimal value. The latter example results in a fairly acceptable simulation but deviations from the target motion are visually obvious.

Using a setup with two boundary clamps, we can apply strong bending deformations and create a “snapping” motion of the real-world specimen. For this motion, we observe a strong increase of the apparent stiffness, suggesting that the non-linearity of the real foam for these deformations is stronger than predicted by the Neohookean model. For the result shown in Fig. 8, the shear modulus is 980 kPa. Similarly, rheometry experiments at higher oscillation frequencies also reach these values around 50 Hz for small strains. Comparing the optimal simulated motion for this example to a motion obtained from material parameters optimized for the slower motion of Fig. 4a shows that these parameters match the deformation reasonably well overall, but fail to damp out some oscillations during the fast “snap”; see Fig. 7 around $t = 0.5$ s. Please see our video for the full motion sample and further comparison to the parameters optimized for the motion in Fig. 4a.

We can also estimate the bulk behaviour of flexible lattice structures as shown in Fig. 5 and 9. As these structures are very complex (the geometry file used for 3D printing the sample in Fig. 5 contains over 130k surface triangles), we want to avoid simulating their dynamics at full resolution. Instead we approximate the geometry in Fig. 5a by a simple bar, meshed with ca. 120 tetrahedral elements. Simulating the motion illustrated in Fig. 5b then takes about 4 s, while the entire optimization finds suitable bulk material parameters in about 5 minutes using 77 individual simulation runs. While the simulation matches the observed oscillation frequency and damping quite well, the RMS error in this case is 10.2 mm. This error is mostly due to a larger deflection orthogonal to the oscillation direction in the simulation. We surmise that the real-world structure is not entirely isotropic, restricting this deflection more than what our isotropic Neohookean model can capture. We leave investigations into anisotropic material models as interesting future work.

Similarly, we only mesh the overall geometry of the hand model in Fig. 9, without representing the lattice structure in simulation. The fingers are made of silicone, 3D printed by *SpectroPlast* [2019]. We again work with a homogeneous, isotropic material in simulation, whereas the real-world specimen clearly has denser regions around the base and the fingertips. Nevertheless, we find material parameters that capture the overall motion behaviour well using this simple simulation model.

Finally, we show an application of our method to soft robotics design in Figure 1. We first fabricate a soft robot prototype from type-III foam using the technique described by [Somm et al. 2019]. Then we capture short motion samples for this particular specimen and estimate material parameters for homogeneous Neohookean elasticity and Newtonian viscosity for a reasonably coarse mesh. Figure 1a shows the collected motion trajectories alongside the best matching simulated ones. We then use the same mesh and parameters, adding some elements representing the weight of the motors, and design a control strategy for this robot in simulation. The ground contact and tendon actuation forces are simulated using the method of [Bern et al. 2019]. The use of an intentionally coarse mesh greatly speeds up motion planning. Furthermore, the foam's material properties vary among different specimens, depending on details of the fabrication process that are difficult to control. After estimating material parameters for this particular specimen and simulation mesh, we successfully design motion control curves using this simulation model. Our robot then performs the designed motion, achieving a satisfactory match between simulation and real-world behaviour as shown in our accompanying video.

Using the adjoint method combined with the LBFGS algorithm, we can optimize a large number of parameters simultaneously. To demonstrate the effectiveness of this approach, we consider a cylinder of 5 cm radius and 2 cm height, fixed to the ceiling on the top, being stretched under gravity. We manually define an objective function that measures the position of the bottom face after a fixed simulated time of 0.2 s. The target position of each point on this face is set to a vertical distance of 2 cm (stretching the cylinder by 100% along its axis) and proportionally offset towards the face's centroid by 20%. In this example, we use a Neohookean elastic and a Newtonian damping model. We first run a global parameter optimization to find the closest solution using homogeneous material parameters. This intermediate solution matches the target reasonably well on average, however, the bottom face must bend as the material drops under gravity, forming a bowl-like shape. Our objective function, on the other hand, asks for this face to remain planar. We now optimize per-element elasticity parameters, starting from the homogeneous solution. As shown in Fig. 10a, the resulting local material parameters form a stiff core in the centre, and soften towards the edges to approach a planar final configuration. We can similarly optimize for local densities instead of local stiffnesses, resulting conversely in a lighter core and a heavier rim as shown in Fig. 10b. Both examples use the same mesh consisting of 8.7k tetrahedral elements, which means that we optimize for 17.4k elastic parameters, or 8.7k density parameters respectively.

In order to test the capabilities of this per-element material parameter estimation method on a real-world scenario, we fabricate a cylinder consisting of type-III (softer) foam on one end and type-V (stiffer) foam on the other. Figures 12 and 13 show results obtained for this specimen. The example shown in Fig. 12 uses a single boundary clamp. In this case, the optimization finds the onset of the softer region, as well as a very stiff region where the physical sample is glued to the rigid clamp (highlighted with a red ellipse in the image). However, for this particular motion, the softer tip is not captured by the optimization; this result is likely due to the fact that only one of seven motion capture markers sits on the very end of the cylinder, which has relatively little influence on the objective function value. Furthermore, the range of stiffness values is unrealistically large.

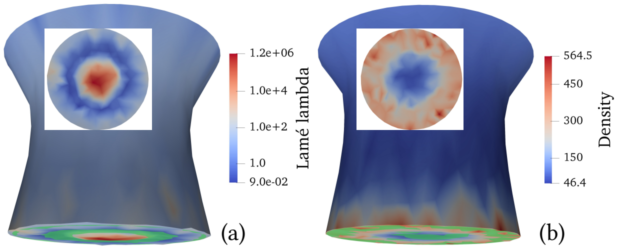

Fig. 10. A cylinder dropping from the ceiling under gravity. The objective function requires the bottom face to match the green disk. We optimize for local (per-element) elastic Lamé parameters (a), or densities (b). Insets show bottom view.

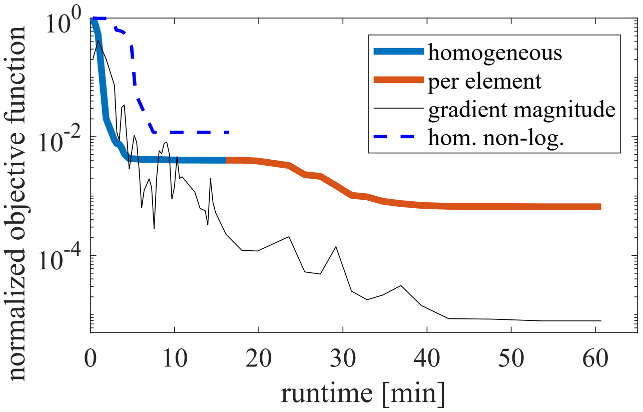

Fig. 11. Convergence of LBFGS optimizing for the logarithm of visco-elastic material parameters. Starting from an initial guess that is too soft, we first optimize globally homogeneous material parameters (blue) and then per-element stiffnesses (red) arriving at the result shown in Fig. 10a. We optimize for the logarithm of the material parameters by default and compare to optimizing for their values directly during the homogeneous part (blue dashes).

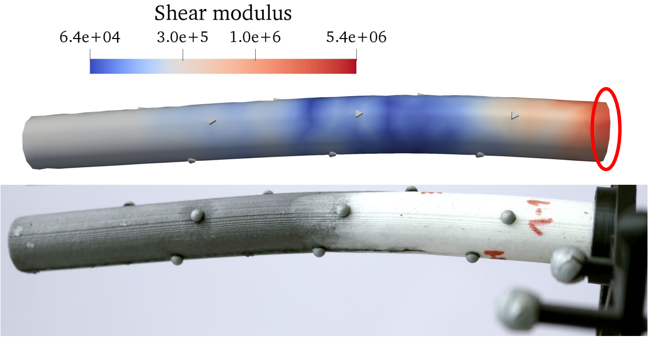

Fig. 12. Optimization result for per-element elasticity (and global viscosity) parameters on a multi-foam cylinder specimen. Single boundary clamp.

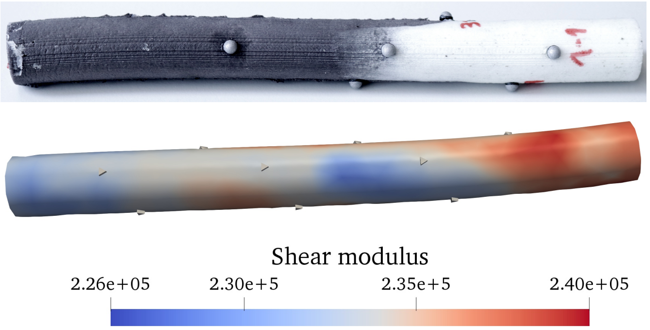

Fig. 13. Optimization result for per-element elasticity (and global viscosity) parameters on a multi-foam cylinder specimen. Two boundary clamps.

In contrast, if we add a second boundary clamp on the other side of the cylinder as well and optimize for the combination of both motions simultaneously, we see a clearer distinction between the softer part on the left of the image and the stiffer part on the right, with a mixed region in between. The overall range of parameters is also more reasonable, see Fig. 13. In this example we use a smoothly rotating motion of the two boundary clamps, with noticeable, but not excessive bending of the cylinder; please also refer to the accompanying video. For this motion, optimizing only for globally homogeneous elastic parameters results in a root-mean-square error per motion capture marker of 6.64 mm, whereas optimizing for local per-element parameters cuts this error to less than half at 3.13 mm (which is less than the diameter of our motion capture markers).

Our tests, as summarized in Figures 3, 6, and 7, show that homogeneous material parameters estimated with our method can generalize to motions that exhibit a similar deformation range and frequency. As our examples mostly capture shear wave propagation behaviour, the material's shear modulus is fitted best. In constrast, optimizing for only an elastostatic snapshot, instead of a dynamic motion, can lead to errors in the shear modulus of over 30% and would not reproduce the desired motion well in simulation (of course damping parameters would need to be determined by some other method in this case).

When optimizing for inhomogeneous parameters on the other hand, we must expect to encounter many local minima. We choose to start from the homogeneous optimal parameters, and apply a gradient based optimization method, which means we expect to find a local minimum near the homogeneous configuration. Nevertheless, using only a single motion sample (Fig. 12) can lead to exaggerated and unrealistic parameter distributions. In this case, combining two motion samples (Fig. 13) improves the result considerably and yields a physically plausible parameter distribution.

While our primary goal is to estimate parameters of well established physical material models, such as Neohookean elasticity, as well as Newtonian or power-law viscous damping, the optimization framework is equally suited to handle more generic material models (so long as partial derivatives of the internal forces can be computed efficiently). To demonstrate this capability, we implemented a principal-stretch-based elastic material as described by Xu et al. [2015]. We use a uniaxial and a volumetric strain-stress response function and initialize the model to a Neohookean initial configuration according to Eq. (25) in [Xu et al. 2015], see also Eq. (19). We then optimize for the spline control point values, while the control point locations in strain space remain constant. As pointed out by Xu et al., a generic spline function can easily lead to a material model that causes unstable simulations unless some precautions are taken. We found that adding a regularization term to our objective function that slightly penalizes negative first derivatives of the splines is sufficient to produce acceptable results. If a simulation becomes unstable during the optimization procedure, we simply return DBL_MAX as objective function value, which causes the line search to backtrack towards the previous (stable) configuration. Figure 14 shows results using 5 and 21 control points per spline respectively. The root-mean-square errors (in terms of distances from the given target) are 0.45 mm and 0.42 mm respectively. In comparison, optimizing for homogeneous material parameters results in an error of 0.81 mm, whereas local elastic parameters (Fig. 10a) achieve a slightly lower RMS error of 0.33 mm, while optimizing for local densities instead results in an error of just 0.08 mm.

As this model still represents a globally homogeneous material, it cannot reach the target as precisely as a per-element optimization; it does, however, achieve a better result than a standard Neohookean model. In our tests we have observed some issues with our implementation of this material model, which if addressed, could further improve results. First, representing the material response curves as interpolating splines can lead to unwanted oscillations of the splines for some control point values, which tend to result in unstable simulations. While the line search ensures stability of the final result, as described above, it also limits the step size and causes the optimization to converge more slowly. Furthermore, without additional constraints, the optimization can introduce residual forces at the rest configuration, which is not physically meaningful. A more carefully constructed parametrization could improve the practical utility of this, or a similar, material model in the future.

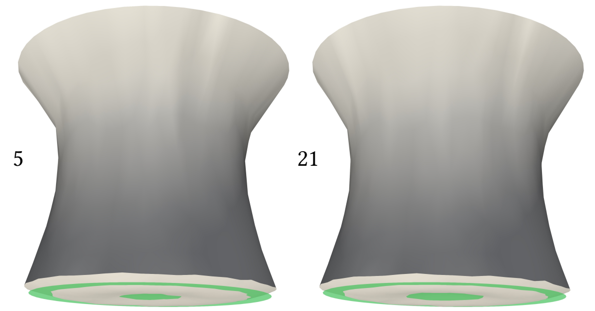

Fig. 14. The same simulation as in Fig. 10, now using a principal-stretch-based elasticity model. We optimize for global spline control point values representing the material's strain-stress response. Image labels specify the number of control points per spline.

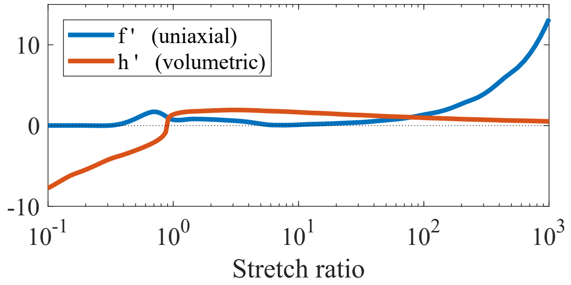

Fig. 15. Resulting spline curves for the principal stretch material using 21 control points per spline, using a uniaxial and a volumetric response function as described in [Xu et al. 2015].

6 DISCUSSION

In this paper we have introduced a new method to estimate visco-elastic material parameters of real-world specimens from dynamic motion data without any force measurements or specialized testing equipment. We build upon a standard finite element simulation and compute objective function gradients analytically through either a direct sensitivity analysis or an adjoint state method. Using a standard LBFGS algorithm, we find the optimal material parameters that allow a chosen simulation model to match the observed motion target in a least-squares sense. We fabricate our specimens by either foam moulding or 3D printing, and we assume that the design geometry coincides with the rest pose of the fabricated objects for our simulations. As we are interested in the viscous damping properties as well as the elastic ones, we use BDF2 time integration (which causes much less numerical damping than the common backward Euler method) in all of our examples. We generally run simulations at the same frame rate as the motion capture system, which is 180 Hz for most examples. As this frame rate is relatively high compared to the stiffness of our specimens, we do not require any addtional stability aids in our FEM framework and use a standard Newton method to solve the non-linear dynamics problem. Of course, any existing FEM implementation including such improvements could be used as well, as long as the derivative terms required for the objective function gradient calculations are provided at the end of each time step.

We have demonstrated the flexibility of our method in various examples, including real-world specimens, custom-designed objective functions or material models, as well as numerical coarsening. The latter example fits a low-resolution simulation to a high-resolution one, Fig. 2, automatically adjusting the simulation parameters to counteract the effects of numerical stiffness and dissipation.

When optimizing for a small number of homogeneous material parameters, the advantage in terms of total run time of the adjoint method is fairly small compared to the direct approach, while the adjoint method consumes more memory. In terms of the resulting gradient, the two methods are equivalent. As most of our examples use reasonably short recorded motion data, we use the adjoint method for most of our results, using less than the available 16 GB of memory. If memory consumption is a concern and the number of parameters is low, or the sensitivies themselves are required for other reasons, direct differentiation should be used instead. We use the direct approach for the homogeneous phase of Fig. 11 for demonstration purposes.

We also show an example using a custom-designed objective function to find a material distribution that achieves a desired deformation behaviour; Fig. 10. Furthermore, we can use more elaborate material models with an arbitrary number of parameters as shown in Fig. 14. However, creating real-world motion samples that result in a generic predictive material model without further assumptions on the type of non-linearity remains an open problem.

Another interesting direction for future work are anisotropic material models. While extending our optimization framework to these models should be relatively straightforward, the question of how to obtain sufficient information about the material's real-world behaviour becomes more challenging to answer. Similarly, any history-dependent material model (such as plasticity) would require us to track further internal state variables and their derivatives in the simulation. We leave such models as an interesting avenue for future research.

Our objective function definition for real-world specimens uses motion capture data including precise timing of the observed behaviour. Consequently, a relatively small, yet visually noticeable, mismatch of the recorded and simulated motion trajectories does not incur as high an objective function penalty as, for instance, a simulation that matches the trajectory well initially but drifts out of phase due to mismatches of the harmonic oscillation frequencies. As a result, we often observe noticeable differences in the trajectories (such as in Fig. 1 and 4). This situation does, however, mean that we do not need to calibrate the marker placement exactly, or worry about errors introduced by the motion capture. Finally, weighting functions emphasizing certain markers, or specific time intervals, during optimization could be introduced into our method. As we aim to present the capabilities of our system without further hyper-parameter tuning we have not implemented such weights in our results.

While we only use parameter dependent (elastostatic) initial conditions in our examples and keep the boundary conditions independent, the presented theory could be extended to cover parameter dependent boundary conditions in an analogous way by considering the derivatives of forces with respect to the parametrization of the boundary data. Similarly, we have not used dynamic re-meshing in our examples. For cases where mesh adaption would be required during the simulation, the sensitivities computed by the direct method can in principle be interpolated from the old to the new mesh alongside all other simulation state data. However, it is currently not clear if the same approach would also work in the adjoint setting, where the adjoint state is integrated backwards in time.

The main application of our system is to estimate physical parameters for complex visco-elastic materials, such as PU foams or 3D printed flexible structured materials. In this way, our method circumvents the need for (a) measuring material parameters with highly specialized testing instruments, then (b) implementing an accurate small-scale simulation model, and finally (c) performing numerical coarsening (or homogenization) resulting in bulk material parameters for fast, coarse simulation models. Instead we directly find the parameters for the coarse simulation. In the end, the overall visco-elastic behaviour of our real-world test specimens is approximated very well by our optimized simulation models.

ACKNOWLEDGMENTS

This work was supported by the ETH Zürich Postdoctoral Fellowship 18-1 FEL-09. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

REFERENCES

Thomas Auzinger, Wolfgang Heidrich, and Bernd Bickel. 2018. Computational design of nanostructural color for additive manufacturing. *ACM Transactions on Graphics* 37, 4 (2018), 1–16. https://doi.org/10.1145/3197517.3201376

Jernej Barbič and Doug L. James. 2005. Real-Time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics* 24, 3 (2005), 982. https://doi.org/10.1145/1073204.1073300

James M. Bern, Pol Banzet, Roi Poranne, and Stelian Coros. 2019. Trajectory optimization for cable-driven soft robot locomotion. *Robotics: Science and Systems* (2019).

B. Bickel, M. Bächer, M. A. Otaduy, W. Matusik, H. Pfister, and M. Gross. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Transactions on Graphics* 28, 3 (2009), 1. https://doi.org/10.1145/1531326.1531395

Javier Bonet and Richard D. Wood. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis.* Cambridge University Press. https://doi.org/10.1017/cbo9780511755446

A. F. Bower, N. A. Fleck, A. Needleman, and N. Ogbonna. 1993. Indentation of a Power Law Creeping Solid. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 441, 1911 (1993), 97–124. https://doi.org/10.1098/rspa.1993.0050

Andrew M. Bradley. 2013. *PDE-constrained optimization and the adjoint method.* Technical Report. Stanford University. https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-aware numerical coarsening for fabrication design. *ACM Transactions on Graphics* 36, 4 (2017), 1–15. https://doi.org/10.1145/3072959.3073669

Desai Chen, David I. W. Levin, Shinjiro Sueda, and Wojciech Matusik. 2015. Data-driven finite elements for geometry and material design. *ACM Transactions on Graphics* 34, 4 (2015), 74:1–74:10. https://doi.org/10.1145/2766889

C.A. Felippa and B. Haugen. 2005. A unified formulation of small-strain corotational finite elements: I. Theory. *Computer Methods in Applied Mechanics and Engineering* 194, 21-24 (2005), 2285–2335. https://doi.org/10.1016/j.cma.2004.07.035

Joachim Georgii and Rüdiger Westermann. 2008. Corotated Finite Elements Made Fast and Stable. https://doi.org/10.2312/pe/vriphys/vriphys08/011-019

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), 131–140. https://doi.org/10.1145/1028523.1028541

Arieh Iserles. 1996. *A First Course in the Numerical Analysis of Differential Equations (Cambridge Texts in Applied Mathematics).* Cambridge University Press.

Doris H. U. Kochanek and Richard H. Bartels. 1984. Interpolating splines with local tension, continuity, and bias control. *ACM SIGGRAPH Computer Graphics* 18, 3 (1984), 33–41. https://doi.org/10.1145/964965.808575

R. Mahnken. 2017. Identification of Material Parameters for Constitutive Equations. *Encyclopedia of Computational Mechanics, Second Edition (eds E. Stein, R. Borst and T. J. Hughes)* (2017). https://doi.org/10.1002/9781119176817.ecm2043

Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid control using the adjoint method. *ACM Transactions on Graphics* 23, 3 (2004), 449. https://doi.org/10.1145/1015706.1015744

E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Computer Graphics Forum* 31 (2012). https://doi.org/10.1111/j.1467-8659.2012.03031.x

Eder Miguel, David Miraut, and Miguel A. Otaduy. 2016. Modeling and Estimation of Energy-Based Hyperelastic Objects. *Computer Graphics Forum* 35, 2 (2016), 385–396. https://doi.org/10.1111/cgf.12840

E. Miguel, R. Tamstorf, D. Bradley, S. C. Schvartzman, B. Thomaszewski, B. Bickel, W. Matusik, S. Marschner, and M. A. Otaduy. 2013. Modeling and estimation of internal friction in cloth. *ACM Transactions on Graphics* 32 (2013).

Matthias Müller and Markus Gross. 2004. Interactive Virtual Materials. In *Proceedings of Graphics Interface 2004.* 239–246.

A.H.W. Ngan, H.T. Wang, B. Tang, and K.Y. Sze. 2005. Correcting power-law viscoelastic effects in elastic modulus measurement using depth-sensing indentation. *International Journal of Solids and Structures* 42, 5-6 (2005), 1831–1846. https://doi.org/10.1016/j.ijsolstr.2004.07.018

Jorge Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Math. Comp.* 35, 151 (1980), 773–773. https://doi.org/10.1090/s0025-5718-1980-0572855-7

James F. O'Brien and Jessica K. Hodgins. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH'99.* ACM Press.

R. W. Ogden. 1997. *Non-Linear Elastic Deformations.* Dover Publications Inc.

W. C. Oliver and G. M. Pharr. 1992. An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments. *Journal of Materials Research* 7, 06 (1992), 1564–1583.

Dinesh K. Pai, Jochen Lang, John Lloyd, and Robert J. Woodham. 1999. ACME, a telerobotic active measurement facility. In *Experimental Robotics VI.* Springer London, 391–400. https://doi.org/10.1007/bfb0119417

Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. 2001. Scanning physical interaction behavior of 3D objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01.* ACM Press.

Yixuan Qiu. 2019. LBFGS++. http://yixuan.cos.name/LBFGSpp. 2019-03-12.

Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids. In *ACM SIGGRAPH 2012 Courses.* ACM Press. https://doi.org/10.1145/2343483.2343501

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Transactions on Graphics* 37, 2 (2018), 1–15. https://doi.org/10.1145/3180491

Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Transactions on Graphics* 38, 1 (2019), 1–15. https://doi.org/10.1145/3241041

Smooth-On, Inc. 2018. FlexFoam-iT!™ Series, Technical Overview. http://www.smooth-on.com/tb/files/FLEXFOAM-IT_SERIES.pdf.

Luca Somm, David Hahn, Nitish Kumar, and Stelian Coros. 2019. Expanding Foam as the Material for Fabrication, Prototyping and Experimental Assessment of Low-Cost Soft Robots With Embedded Sensing. *IEEE Robotics and Automation Letters* 4, 2 (2019), 761–768. https://doi.org/10.1109/lra.2019.2893416

SpectroPlast. 2019. https://spectroplast.com/.

Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. Energetically Consistent Invertible Elasticity. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12).* 25–32.

Stratasys, Ltd. 2018. The Stratasys F123 Series. http://www.stratasys.com/-/media/files/printer-spec-sheets/pss_fdm_f123series_1017a.pdf

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* ACM Press. https://doi.org/10.1145/1073368.1073394

Rosell Torres, Alejandro Rodríiguez, José M. Espadero, and Miguel A. Otaduy. 2016. High-resolution interaction with corotational coarsening models. *ACM Transactions on Graphics* 35, 6 (2016), 1–11. https://doi.org/10.1145/2980179.2982414

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation capture and modeling of soft objects. *ACM Transactions on Graphics* 34, 4 (2015), 94:1–94:12. https://doi.org/10.1145/2766911

Chris Wojtan, Peter J. Mucha, and Greg Turk. 2006. Keyframe Control of Complex Particle Systems Using the Adjoint Method. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06).* Eurographics Association, 15–23. http://dl.acm.org/citation.cfm?id=1218064.1218067

Hongyi Xu and Jernej Barbič. 2017. Example-based damping design. *ACM Transactions on Graphics* 36, 4 (2017), 1–14. https://doi.org/10.1145/3072959.3073631

Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbič. 2015. Nonlinear material design using principal stretches. *ACM Transactions on Graphics* 34, 4 (2015), 75:1–75:11. https://doi.org/10.1145/2766917