



HAL
open science

Finite element solver for data-driven finite strain elasticity

Auriane Platzzer, Adrien Leygue, Laurent Stainier, Michael Ortiz

► **To cite this version:**

Auriane Platzzer, Adrien Leygue, Laurent Stainier, Michael Ortiz. Finite element solver for data-driven finite strain elasticity. *Computer Methods in Applied Mechanics and Engineering*, 2021, 379, 10.1016/j.cma.2021.113756 . hal-03436816

HAL Id: hal-03436816

<https://hal.science/hal-03436816v1>

Submitted on 1 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Finite element solver for data-driven finite strain elasticity

Auriane Platzzer^a, Adrien Leygue^a, Laurent Stainier^{a,*}, Michael Ortiz^b

^a*Institut de Recherche en Génie Civil et Mécanique (GeM), UMR 6183, CNRS - École Centrale de Nantes
- Université de Nantes, 1, rue de la Noë, 44321 Nantes, France*

^b*Graduate Aerospace Laboratories, California Institute of Technology, 1200 E. California Blvd., Pasadena,
CA, 91125, USA*

Abstract

A nominal finite element solver is proposed for data-driven finite strain elasticity. It bypasses the need for a constitutive model by considering a database of deformation gradient/first Piola-Kirchhoff stress tensors pairs. The boundary value problem is reformulated as the constrained minimization problem of the distance between (i) the mechanical states, *i.e.* strain-stress, in the body and (ii) the material states coming from the database. The corresponding constraints are of two types: kinematical, *i.e.* displacement-strain relation, and mechanical, *i.e.* conservation linear and angular momenta. The solver uses alternated minimization: the material states are determined from a local search in the database using an efficient tree-based nearest neighbor search algorithm, and the mechanical states result from a standard constrained minimization addressed with an augmented Lagrangian approach. The performance of the solver is demonstrated by means of 2D sanity check examples: the data-driven solution converges to the classical finite element solution when the material database increasingly approximates the constitutive model. In addition, we demonstrate that the balance of angular momentum, which was classically not taken into account in previous data-driven studies, must be enforced as a constraint to ensure the convergence of the method.

Keywords: Data-driven computing; finite strain; model-free; optimization methods; data science.

1. Introduction

Computational mechanics is a field in which a large amount of *data* is both consumed and produced. On the one side, the recent developments of experimental measurements techniques have provided rich databases for the identification process of constitutive models [1, 2]. On the other side, heavy computations, such as multi-scale analyses, produce a huge amount of discrete values of **kinematical and stress quantities**, which can be used in the end to extract knowledge on the overall material behavior. A typical entry-point for *Data Science*

*Corresponding author.

Email address: laurent.stainier@ec-nantes.fr (Laurent Stainier)

in the field of computational mechanics is then material modeling. For instance, artificial neural networks [3, 4, 5], applied to computational homogenization of nonlinear elasticity at small strain [6] and large strain [7], or manifold learning techniques [8] have been considered to find underlying structure in the data. In these approaches however, some sort of material modeling is still present, and the data is discarded at the end of the process. In contrast, [9] recently introduced the data-driven computational mechanics (DDCM) paradigm, which incorporates data *directly* into the computation. This pioneer work was consolidated by the mathematical framework proposed in [10], demonstrating that DDCM encompasses the classical definition of the boundary value problem (BVP) in solid mechanics.

1.1. State of the art

The DDCM approach relies on the fundamental separation between the characterization of the material response and the satisfaction of essential constraints and conservation laws: while the former is most likely to be approximately captured (*e.g.* due to noise in the acquisition of experimental data), the latter ones must be exactly satisfied. In solid mechanics, the typical constitutive variables involved are the strain and the stress. The BVP is therefore reformulated as a minimization of a distance function between two types of strain-stress fields: one, representing the material response, takes value in a database resulting from experimental acquisition of material data points, and the other, verifying both compatibility and equilibrium equations, represents the admissible mechanical state of the body. The former strain-stress field is referred to as the *material state* and belongs to a *material data set*, while the latter *mechanical state* belongs to a *constraint set*. Both sets are subspaces of the *phase space*, which collects all possible strain-stress fields.

The DDCM paradigm was introduced prior to the actual availability of experimental data points: if the strain tensor field can be obtained from full-field measurements, the stress field is often only measured in specific one-dimensional experimental tests. Since then, recent works have paved the way for the use of experimental databases in data-driven engineering: initially formulated by [11] as a non-parametric inverse approach inspired by DDCM, the so-called data-driven identification (DDI) technique identifies stress fields from full-field measurements of strains and applied resultant forces, without any constitutive equation. It has been shown to be of particular interest for supplying DDCM simulations with importance sampled material data sets [12]. The inverse method was also developed for elasto-plasticity by [13] and further explored in nonlinear elasticity, plasticity and elasto-dynamics in [14]. It was also successfully extended to large strain and assessed with *raw* experimental data in [15], which provides encouraging prospects. Alternatively, the DDCM can also be employed with numerically generated databases from multiple simulations at finer scales [16, 17]. Yet, in the current stages of development, including the present work, the use of *in silico* data, generated by a sampling of a constitutive model later discarded in the data-driven simulations, allows to assess the approach by comparing to standard finite element (FE) analyses.

Recently, several authors contributed to improve the DDCM numerical scheme: the entropy-maximizing solver developed in [18] is more robust to noisy material data sets with

outliers; the issue of the high dimensionality of the phase space was addressed by [19] using tensor voting, a machine learning technique; [20] showed that the data-driven BVP is a well posed Mixed-Integer Quadratic Programming problem, allowing to reach the global minimizer, through (very expensive) branch-and-bound solvers. *Hybrid* methods have also been introduced to further improve the robustness of the data-driven approach: bridging the gap between *inverse* manifold reconstruction techniques [8] and *direct* DDCM solvers, several authors have incorporated the definition of a constitutive manifold into the data-driven distance minimizing problem [21, 22, 23, 24]. For instance, [23, 24] formulated an approximate nonlinear optimization problem and demonstrated its computational efficiency in the static and dynamic cases for specific configurations spaces of nonlinear kinematics. Finally, the DDCM has been extended to other classes of problem, such as elasto-dynamics [25], diffusion problems [26], history-dependent behavior like visco-elasticity [27] or fracture mechanics [28].

1.2. Aim of the present work

Extending the DDCM supposes to choose the relevant pair of constitutive variables to work with, *i.e.* to define the appropriate phase space of the problem. An extension of particular interest is finite strain elasticity, for which there exist several strain-stress work conjugate pairs. To the best of our knowledge, this includes two contributions so far in **solid mechanics**: the initial work of [29] proposed a phase-space of Green-Lagrange strain–second Piola-Kirchhoff stress pairs, while [30] extended their small strain framework [10] using deformation gradient–first Piola-Kirchhoff stress tensor pairs¹. In the following, we refer to the former as the *Lagrangian* formulation and to the latter as the *nominal* formulation. In a previous work [31], we consolidated the finite element formulation of [29] and assessed the approach with synthetic data. The work of [30] provides mathematical ground for convergence of the data-driven solution towards the classical solution using the nominal framework; we could not find however a similar analysis for the Lagrangian formulation in the literature. To compare both formulations, the nominal continuum mechanics formulation of [30] still needs to be implemented in a generic finite element setting. In addition, the choice of phase space is crucial to deal with the conservation of angular momentum: in the Lagrangian formulation, the symmetry of the phase space tensors readily ensured it; its enforcement in the nominal formulation is a topic addressed in this paper.

The main objective of the present work is then **two fold**: (i) to provide a FE data-driven solver for the nominal formulation, **inspired by** the Lagrangian FE solver presented in [31]; (ii) to assess whether momentum balance should be encoded in the constraint set or if verifying the principle in the material data set is enough. As a secondary contribution, we also give some practical advice on the use of finite strain data-driven solvers. In this paper, we deliberately detail the equations and algorithms as much as possible in a concern of reproducible and open science.

¹We distinguished these contributions from the Eulerian framework adopted for large strain DDI by [15]: the logarithmic strain–Cauchy stress formulation used therein has not been used in a DDCM setting yet.

The paper is organized as follows. In Section 2, we develop a finite element solver for the nominal formulation of data-driven finite strain elasticity. The solver is further evaluated in Section 3, with three different test cases, of increasing complexity. All are two-dimensional problems, assuming plane stress conditions and incompressibility of the material. The data-driven results are compared with the reference solution, obtained from classical FE analysis. Finally, we summarize the key features of this work and discuss some of their implications in Section 4.

2. Development of a nominal finite strain data-driven solver

The nominal formulation has first been introduced by [30] to extend the data-driven framework defined in [10] to finite strain elasticity. The phase space consists of deformation gradient–first Piola-Kirchhoff stress tensor fields (\mathbf{F}, \mathbf{P}) . Hence, the local phase space can no longer be reduced to $\mathbb{R}_{\text{sym}}^{n \times n} \times \mathbb{R}_{\text{sym}}^{n \times n}$, which directly enforces material-frame indifference and conservation of angular momentum in geometrically linear elasticity and Lagrangian finite strain elasticity. In [30], the former is encoded in the material data set by using an orbit representation: if a material data point (\mathbf{F}, \mathbf{P}) is in the database, then so is its orbit $(\mathbf{QF}, \mathbf{QP})$, $\mathbf{Q} \in SO(n)$ under the left action of the Special Orthogonal group $SO(n) = \{\mathbf{Q} \in \mathbb{R}^{n \times n} \mid \mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}\}$. The latter can be treated as an additional constraint to the minimization problem, as explored in the following.

This section is dedicated to recalling the continuum mechanics formulation as introduced by [30] and developing a FE data-driven solver on that basis, analogous to the one derived in [31] for the Lagrangian formulation.

2.1. Continuum mechanics

Let us consider an elastic body occupying a domain $\Omega_0 \in \mathbb{R}^n$ in the reference configuration and $\Omega \in \mathbb{R}^n$ under quasi-static loading, in dimension n . For now, the general three-dimensional case $n = 3$ is considered only; specific assumptions allowing for the reduction to two-dimensional mechanics are discussed next. A material point has \mathbf{X} and \mathbf{x} as initial and current coordinates, respectively. The deformation-stress state $(\mathbf{F}(\mathbf{X}), \mathbf{P}(\mathbf{X}))$ of a material point \mathbf{X} is referred to as the *local state* of \mathbf{X} . The *local state function* z is then defined as the mapping between every material point \mathbf{X} of the reference domain Ω_0 to its local state:

$$\begin{aligned} z : \quad \Omega_0 &\mapsto \mathcal{Z}_{\text{loc}} \\ \mathbf{X} &\rightarrow z(\mathbf{X}) = (\mathbf{F}(\mathbf{X}), \mathbf{P}(\mathbf{X})) \end{aligned} \tag{1}$$

where the local phase space \mathcal{Z}_{loc} is $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$. The global phase space \mathcal{Z} is the collection of local state functions $z : \Omega_0 \rightarrow \mathcal{Z}_{\text{loc}}$. As shown in [30], to ensure convergence properties, the deformation gradient and the stress fields must each belong to a given L^p space, *i.e.* they must be measurable functions for which the p -th power of the absolute value is Lebesgue integrable. The global phase space is then defined as

$$\mathcal{Z}_{p,q}(\Omega_0) = L^p(\Omega_0, \mathbb{R}^{n \times n}) \times L^q(\Omega_0, \mathbb{R}^{n \times n}) \tag{2}$$

with $(p, q) \in (1, \infty)$ such that $1/p + 1/q = 1$. As aforementioned, the local phase space no longer consists of symmetric tensors, as compared with the small strain or Lagrangian finite strain settings. Material frame-indifference and conservation of angular momentum constraints must then be enforced specifically.

The material data set \mathcal{D} is defined by the collection of local state functions mapping to nominal strain-stress pairs of a local material database \mathcal{D}_{loc} :

$$\mathcal{D} = \{z \in \mathcal{Z} \mid z(\mathbf{X}) \in \mathcal{D}_{\text{loc}}\}, \quad (3a)$$

with

$$\mathcal{D}_{\text{loc}} = \{(\mathbf{F}_i, \mathbf{P}_i), i \in [1 \dots m]\} \subset \mathcal{Z}_{\text{loc}}, \quad (3b)$$

with m the number of material data points.

The constraint set \mathcal{E} consists of local state functions verifying mechanical admissibility. It comprises three constraints, namely

(i) the kinematics constraints, defined as

$$\mathbf{F}(\mathbf{X}) = \nabla_0 \mathbf{u}(\mathbf{X}) + \mathbf{I} \quad \text{in } \Omega_0, \quad (4a)$$

$$\mathbf{u}(\mathbf{X}) = \mathbf{u}_D(\mathbf{X}) \quad \text{on } \Gamma_{0_D}, \quad (4b)$$

(ii) the translational equilibrium, written as

$$\text{DIV } \mathbf{P}(\mathbf{X}) + \mathbf{f}_0(\mathbf{X}) = \mathbf{0} \quad \text{in } \Omega_0, \quad (5a)$$

$$\mathbf{P}(\mathbf{X})\mathbf{N}(\mathbf{X}) = \mathbf{t}_0(\mathbf{X}) \quad \text{on } \Gamma_{0_N}, \quad (5b)$$

(iii) the rotational equilibrium, written as

$$\mathbf{F}\mathbf{P}^T = \mathbf{P}\mathbf{F}^T, \text{ in } \Omega_0, \quad (6)$$

where \mathbf{u}_D is the prescribed displacement field on the Dirichlet boundary $\Gamma_{0_D} \subset \partial\Omega_0$, \mathbf{I} is the second order identity tensor, $\nabla_0 = \partial/\partial\mathbf{X}$ and DIV denote the gradient and divergence operators with respect to the initial coordinates, respectively, \mathbf{f}_0 are the body forces per unit reference volume, and \mathbf{t}_0 is the traction vector per unit reference area, acting on the Neumann reference boundary Γ_{0_N} with outer normal \mathbf{N} . The set \mathcal{E} then writes

$$\mathcal{E} = \{z \in \mathcal{Z} \mid (4), (5) \text{ and } (6)\}. \quad (7)$$

Note that the first two constraints (4) and (5) are linear in \mathbf{u} and \mathbf{P} respectively while the rotational equilibrium (6) is a bi-linear constraint and more difficult to enforce in a discrete setting. It can then be more convenient to work with the affine subspace $\mathcal{E}_0 \subset \mathcal{E}$, defined as the collection of local state functions verifying compatibility and translational equilibrium (4) and (5) only.

For later purposes, the translational equilibrium and attached Neumann conditions (5) are reformulated in their weak form. The nominal formulation of the principle of virtual work writes:

$$\delta W = \int_{\Omega_0} \mathbf{P} : \delta \mathbf{F} \, d\mathbf{X} - \int_{\Omega_0} \mathbf{f}_0 \cdot \delta \boldsymbol{\phi} \, d\mathbf{X} - \int_{\partial\Omega_0} \mathbf{t}_0 \cdot \delta \boldsymbol{\phi} \, dA = 0, \quad (8)$$

for any arbitrary kinematically admissible virtual displacement $\delta \boldsymbol{\phi}$ from the current position of the body, where we used the notation $\delta \mathbf{F} = \nabla_0 \delta \boldsymbol{\phi}$. In addition, the conservation of angular momentum (6) is more conveniently expressed by the vector equation

$$\boldsymbol{\epsilon} : (\mathbf{F}(\mathbf{X}) \mathbf{P}^T(\mathbf{X})) = \mathbf{0}, \quad \forall \mathbf{X} \in \Omega_0, \quad (9)$$

where $\boldsymbol{\epsilon}$ denotes the Levi-Civita third-order tensor ($\epsilon_{ijk} = 1$ if $\{i, j, k\}$ is an even permutation of $\{1, 2, 3\}$, -1 if it is odd, and 0 if any indices are repeated). Note that (6) simply means $\mathbf{S} = \mathbf{S}^T$, with $\mathbf{S} = \mathbf{F}^{-1} \mathbf{P}$ the second Piola-Kirchhoff stress tensor used in the Lagrangian formulation, since $\boldsymbol{\epsilon} : \mathbf{A} = \mathbf{0}$ enforces the symmetry condition on \mathbf{A} .

The solution set \mathcal{S} of the data-driven BVP corresponds to the set of local state functions verifying mechanical admissibility meanwhile being closest to the material data set. It then results from the double minimization problem

$$\min_{z \in \mathcal{E}} \min_{z' \in \mathcal{D}} d(z, z') \quad (10)$$

as

$$\mathcal{S} = \arg \min_{z \in \mathcal{E}} \min_{z' \in \mathcal{D}} d(z, z'), \quad (11)$$

with d an appropriate deviation or distance function from the material data set. The deviation function proposed by [30] is

$$d(z, z') = \int_{\Omega_0} (V(\mathbf{F}(\mathbf{X}) - \mathbf{F}'(\mathbf{X})) + V^*(\mathbf{P}(\mathbf{X}) - \mathbf{P}'(\mathbf{X}))) \, d\mathbf{X}, \quad (12)$$

$$\forall (z : \mathbf{X} \mapsto (\mathbf{F}, \mathbf{P}), z' : \mathbf{X} \mapsto (\mathbf{F}', \mathbf{P}')) \in \mathcal{Z} \times \mathcal{Z}.$$

where V is a convex function, with convex conjugate V^* . An appropriate choice for V and V^* is [30, Assumption 2.15]:

$$V(\mathbf{F}) = C^{p/2} \frac{1}{p} |\mathbf{F}|^p = \frac{C^{p/2}}{p} (\text{tr } \mathbf{F}^T \mathbf{F})^{p/2}, \quad (13a)$$

$$V^*(\mathbf{P}) = C^{-q/2} \frac{1}{q} |\mathbf{P}|^q = \frac{C^{-q/2}}{q} (\text{tr } \mathbf{P}^T \mathbf{P})^{q/2}, \quad (13b)$$

where C is a scaling parameter between deformation gradient and stress values which we added to the original definitions of V and V^* in [30]. Herein, C is defined as a scalar, which would correspond to the modulus of a fourth-order tensor.

The framework derived by [30], and recalled above, ensures the existence of so-called *classical solutions* to the minimization problem, provided specific conditions are verified by

the local material data set \mathcal{D}_{loc} . A classical solution is met when the mechanical state field $z \in \mathcal{E}$ and the material state field $z' \in \mathcal{D}$ coincide such that the minimum of $d(z, z')$ is 0. This typically arises in cases where the material data set is achieved from a hyperelastic potential, **but not exclusively**. For more details on the conditions of existence of classical solutions and examples of such material data sets, the reader is referred to [30].

2.2. Remark on two-dimensional problems

Before moving to the finite element discretization of the finite strain data-driven setting presented above, we discuss the reduction to two-dimensional problems.

Two-dimensional problems typically arise under plane strain or plane stress conditions. We show in this section that the local phase space \mathcal{Z}_{loc} is safely reduced to $\mathbb{R}^{2 \times 2} \times \mathbb{R}^{2 \times 2}$ in both cases.

On the one hand, let us consider a BVP **where** a plane stress condition is verified along \mathbf{e}_3 and the material is incompressible. The former constraint enables the stress tensor $\mathbf{P} \in \mathbb{R}^{3 \times 3}$ to be reduced to its in-plane components $\mathbf{P}_{(2)} \in \mathbb{R}^{2 \times 2}$ only, as

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{(2)} & \mathbf{0} \\ \mathbf{0} & P_{33} \end{pmatrix}_{(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)}, \quad (14)$$

with $P_{33} = 0$. Together with the plane stress condition $\mathbf{P}\mathbf{e}_3 = 0$, the latter constraint enables to also only retain the in-plane components $\mathbf{F}_{(2)}$ of the deformation gradient tensor $\mathbf{F} \in \mathbb{R}^{3 \times 3}$, as $J = \det \mathbf{F} = 1$ and

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{(2)} & \mathbf{0} \\ \mathbf{0} & F_{33} \end{pmatrix}_{(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)}, \quad (15)$$

where the out-of-plane stretch ratio $F_{33} = 1/\det \mathbf{F}_{(2)}$ can be determined *a posteriori*.

On the other hand, when plane strain conditions are assumed, $F_{33} = 1$ and $\mathbf{F}_{(2)}$ suffices to characterize the deformation state. However, the out-of-plane stress P_{33} can not be determined in the data-driven approach, as no constitutive equation relates the two three-dimensional tensors. The DD results of a plane strain problem will then be purely two-dimensional.

In summary, provided that the material data set is constituted with two-dimensional deformation gradient–first Piola–Kirchhoff stress tensors, the data-driven approach derived above is unchanged whether $n = 2$ or $n = 3$. Note that a similar rationale allows for the definition of data-driven one-dimensional settings, with applications to other types of mechanical assumption (*e.g.* beam theory).

2.3. Finite element formulation

In [30], the existence of *generalized* data-driven solutions, with $z \neq z'$, is not addressed. These solutions are expected when the local material data set is discrete, *i.e.* contains a finite number of material data points. **As mentioned in the Introduction, such** data sets can arise from experiments (*e.g.* using the DDI technique), numerical simulations at finer scales

or be generated synthetically by sampling of a hyperelastic constitutive model. In this work, we used the latter method as a first assessment of the data-driven approach. To evaluate the corresponding data-driven solutions, we derive in the following a finite element formulation of the data-driven BVP formulated by [30]. We must emphasize that to date, and to the best of our knowledge, **such a derivation is not found in the literature**.

2.3.1. Preliminary considerations

In principle, the values of (p, q) and C **in the deviation functions (13)** could be optimized as well, as their optimal values depend on the BVP and the material data set (see examples provided in [30]). In practice, taking C as the **modulus of the linearized stiffness tensor** of the material (when $\mathbf{F} = \mathbf{I}$) was found to be a reasonable choice. As for the values of (p, q) , note that the functions V and V^* are not differentiable at $\mathbf{0} \in \mathbb{R}^{n \times n}$, except when $(p, q) = (2, 2)$. In addition, we wanted to use the same tree-based algorithm as the one mentioned in [31] to search the material database: our experience shows that these algorithms considerably improve the computational cost of the search for optimal material states, **as compared to a linear search**. However, we could not derive the required pre-processing of data points for values other than $(p, q) = (2, 2)$. The nominal finite element solver presented in the following is then restricted to $p = q = 2$. Thus, the deviation functions used in the FE implementation are

$$V(\mathbf{F}) = \frac{1}{2} C |\mathbf{F}|^2, \text{ and } V^*(\mathbf{P}) = \frac{1}{2} C^{-1} |\mathbf{P}|^2. \quad (16)$$

More details on the tree-based search approach used herein are given in the next section.

2.3.2. Discretization of local state fields

To transition from continuum mechanics equations to a FE setting similar to the Lagrangian formulation presented in [31], we explicitly derive the interpolation functions for the deformation gradient field and the integration rule on a FE mesh.

To this end, let us consider a FE mesh of the body in Ω_0 , consisting of N nodes, labeled $a \in [1 \dots N]$. The elements are equipped with a quadrature rule such that every integral of a function f over Ω_0 is evaluated as the weighted sum of the function values at the integration points:

$$\int_{\Omega_0} f(\mathbf{X}) \, d\mathbf{X} = \sum_{e=1}^M w_e f(\mathbf{X}_e), \quad (17)$$

where M is the total number of integration points in the mesh, labeled $e \in [1 \dots M]$, and w_e is the weight associated to the integration point of coordinate \mathbf{X}_e in the reference configuration Ω_0 . In addition, let $\alpha \in [1 \dots N \times n]$ label the degrees of freedom; note that here²

$$\alpha = \alpha(a, i) = a \times (n - 1) + i \quad (18)$$

²as one possibility among others.

with $i \in [1 \dots n]$ the direction. The displacement field is interpolated in Ω_0 in terms of nodal values $\mathbf{u} = \{\mathbf{u}_a\}_{a=1}^N$ and nodal shape functions $\{\mathcal{N}_a\}_{a=1}^N$ as

$$\mathbf{u}(\mathbf{X}) = \sum_{a=1}^N \mathcal{N}_a(\mathbf{X}) \mathbf{u}_a, \forall \mathbf{X} \in \Omega_0. \quad (19)$$

From (4a), the approximate deformation gradient then writes

$$\mathbf{F}(\mathbf{X}) = \nabla_0 \mathbf{u}(\mathbf{X}) + \mathbf{I}, \text{ with } \nabla_0 \mathbf{u}(\mathbf{X}) = \sum_{a=1}^N (\mathbf{u}_a \otimes \nabla_0 \mathcal{N}_a(\mathbf{X})), \forall \mathbf{X} \in \Omega_0. \quad (20)$$

In the standard FE method, the approximation of the stress field readily results from the approximation of the displacement field, *via* the compatibility and the constitutive equations. In the data-driven approach however, the mechanical stress field \mathbf{P} verifying equilibrium (8) and (9) is no longer directly related to the displacement field. The definition of the functional space for the FE stress field in the data-driven setting is out of the scope of the present work. Instead, the stresses are merely evaluated at the integration points, using (17), as

$$\int_{\Omega_0} \mathbf{P}(\mathbf{X}) \, d\mathbf{X} = \sum_{e=1}^M w_e \mathbf{P}_e \quad (21)$$

where $\mathbf{P}_e = \mathbf{P}(\mathbf{X}_e)$, with \mathbf{X}_e the coordinate of integration point e in Ω_0 .

2.3.3. Phase space and subspaces in the finite element setting

We can now formulate the data-driven BVP in the FE setting. First, we derive the FE definitions of the global phase space \mathcal{Z} and deviation function d . Second, the material data set \mathcal{D} is defined as the corresponding discrete subset of \mathcal{Z} . Third, we inject the evaluation of the deformation gradient and stress fields into the kinematics and equilibrium constraints (4), (8) and (9) in order to express the constraint set \mathcal{E} .

1. The FE local state function z_e maps integration point e in the reference configuration to the corresponding local state pair:

$$\begin{aligned} z_e : \quad \Omega_0 &\rightarrow \mathcal{Z}_{\text{loc}}^e = \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \\ \mathbf{X}_e &\mapsto (\mathbf{F}_e, \mathbf{P}_e) = (\mathbf{F}(\mathbf{X}_e), \mathbf{P}(\mathbf{X}_e)), \end{aligned} \quad (22)$$

The global phase space being the product set $\mathcal{Z} = \mathcal{Z}_{\text{loc}}^1 \times \dots \times \mathcal{Z}_{\text{loc}}^M$, a global point z in \mathcal{Z} is the collection of M local state functions: $z = \{z_e\}_{e=1}^M$. The global deviation function (12) is then evaluated on the discrete mesh as

$$\begin{aligned} d(z, z') &= \sum_{e=1}^M w_e (V(\mathbf{F}_e - \mathbf{F}'_e) + V^*(\mathbf{P}_e - \mathbf{P}'_e)), \\ &\forall \left(z = \{z_e = (\mathbf{F}_e, \mathbf{P}_e)\}_{e=1}^M, z' = \{z'_e = (\mathbf{F}'_e, \mathbf{P}'_e)\}_{e=1}^M \right) \in \mathcal{Z} \times \mathcal{Z}, \end{aligned} \quad (23)$$

with V and V^* as in (16).

2. The material data set $\mathcal{D} \subset \mathcal{Z}$ for the finite element problem is easily derived. For each integration point e , a local material data set $\mathcal{D}_{\text{loc}}^e$ comprises m_e deformation gradient-stress pairs and the global material data set is the collection of local state functions mapping every integration point to a material data point in $\mathcal{D}_{\text{loc}}^e$:

$$\mathcal{D} = \{z \in \mathcal{Z} \mid \forall e \in [1 \dots M], z_e(\mathbf{X}_e) \in \mathcal{D}_{\text{loc}}^e\}, \quad (24a)$$

with

$$\mathcal{D}_{\text{loc}}^e = \{(\mathbf{F}_i, \mathbf{P}_i) \mid i \in [1 \dots m_e]\} \subset \mathcal{Z}_{\text{loc}}^e. \quad (24b)$$

This formulation allows for taking into account various materials in the structure. In the rest of the paper, the terms *material database* are used to refer to local material data sets $\mathcal{D}_{\text{loc}}^e$, whereas the material data set refers to the functional set \mathcal{D} .

3. The Dirichlet boundary conditions are encoded in an array u^{D} of n_{D} scalar prescribed displacements $u^{\text{D}} = \{u_{\alpha}^{\text{D}}\}_{\alpha \in D}$ with $D = \{\alpha_1, \dots, \alpha_{n_{\text{D}}}\}$, such that $\alpha_k \in [1 \dots Nn]$, $\forall k \in [1 \dots n_{\text{D}}]$. The kinematic constraints (4) now write as the discrete form of the boundary conditions, together with the interpolation of the deformation gradient field (20):

$$\mathbf{F}_e = \nabla_0^e \mathbf{u} + \mathbf{I}, \quad \forall e \in [1 \dots M], \quad (25a)$$

$$\mathbf{u}_a \cdot \mathbf{e}_i = u_{\alpha}^{\text{D}}, \quad \forall (a, i) : \alpha(a, i) \in D, \quad (25b)$$

with $\nabla_0^e \mathbf{u} = \nabla_0 \mathbf{u}(\mathbf{X}_e)$ from (20). Using the same interpolation for $\delta \phi$ as for the displacement field (19), and the integration rule (17), the discrete form of (8) writes

$$\delta \phi_a \cdot \left(\sum_{e=1}^M w_e \mathbf{P}_e \mathbf{B}_{ea} - \mathbf{T}_a^{\text{ext}} \right) = 0, \quad \forall a \in [1 \dots N], \quad (26)$$

with the vector $\mathbf{B}_{ea} = \nabla_0 \mathcal{N}_a(\mathbf{X}_e)$ and $\mathbf{T}_a^{\text{ext}}$ the external force vector defined as

$$\mathbf{T}_a^{\text{ext}} = \int_{\Omega_0} \mathcal{N}_a \mathbf{f}_0 \, d\mathbf{X} + \int_{\partial\Omega_0} \mathcal{N}_a \mathbf{t}_0 \, dA. \quad (27)$$

The discrete form of the translational equilibrium is then defined as the nullity of the discretized residual force vector $\mathbf{R} = \{\mathbf{R}_a\}_{a=1}^N$ on the degrees of freedom $\alpha \notin D$:

$$\mathbf{R}_a \cdot \mathbf{e}_i = 0, \quad \forall (a, i) : \alpha(a, i) \notin D, \quad (28a)$$

with

$$\mathbf{R}_a = \sum_{e=1}^M w_e \mathbf{P}_e \mathbf{B}_{ea} - \mathbf{T}_a^{\text{ext}}, \quad \forall a \in [1 \dots N]. \quad (28b)$$

As for the conservation of angular momentum, (9) holds at every integration point. Thus, the third mechanical constraint simply writes

$$\mathbf{r}_e = \boldsymbol{\epsilon} : (\mathbf{F}_e \mathbf{P}_e^T) = \mathbf{0}, \quad \forall e \in [1 \dots M]. \quad (29)$$

The set of mechanically admissible FE local state functions is then

$$\mathcal{E} = \{z \in \mathcal{Z} \mid (25), (28) \text{ and } (29)\}. \quad (30)$$

Finally, from (11) and (23), the FE nominal data-driven BVP is also expressed as a double minimization problem

$$\mathcal{S} = \arg \min_{z \in \mathcal{E}} \min_{z' \in \mathcal{D}} \sum_{e=1}^M w_e d_e(z_e, z'_e) \quad (31)$$

where d_e denotes the local deviation function in $\mathcal{Z}_{\text{loc}}^e \times \mathcal{D}_{\text{loc}}^e$, defined as

$$d_e(z_e, z_i) = V(\mathbf{F}_e - \mathbf{F}_i) + V^*(\mathbf{P}_e - \mathbf{P}_i), \quad \forall e \in [1 \dots M], \forall z_e = (\mathbf{F}_e, \mathbf{P}_e) \in \mathcal{Z}_{\text{loc}}^e, \forall z_i = (\mathbf{F}_i, \mathbf{P}_i) \in \mathcal{D}_{\text{loc}}^e, \quad (32)$$

with V and V^* as in (16).

2.4. Alternated minimization

As the other data-driven systems that have been developed so far, the nominal formulation yields a minimization problem coupled with a combinatorial optimization. For the sake of simplicity, our first attempt for a nominal finite strain data-driven solver also makes use of the alternated minimization introduced by [9], and used in [29, 31]. The data-driven problem (31) is then split into two distinct problems:

- Let us first consider the stage where the mechanical state $z = \{z_e = (\mathbf{F}_e, \mathbf{P}_e)\}_{e=1}^M \in \mathcal{E}$ is fixed. The material solution set \mathcal{S}^{mat} is then determined from (31) as

$$\mathcal{S}^{\text{mat}} = \arg \min_{z' \in \mathcal{D}} \sum_{e=1}^M w_e d_e(z_e, z'_e). \quad (33)$$

Note that $\mathcal{D} = \mathcal{D}_{\text{loc}}^1 \times \dots \times \mathcal{D}_{\text{loc}}^M$ is a product set of very high dimension ($\sum_{e=1}^M m_e$) which makes the minimization problem (33) *a priori* intractable. However, as all terms in the summation are independent from one another, the minimization can be conducted in each material database $\mathcal{D}_{\text{loc}}^e$ separately. The material solution set \mathcal{S}^{mat} then results from M searches for nearest neighbors in local phase space:

$$\mathcal{S}^{\text{mat}}(z) = \left\{ \arg \min_{z_i \in \mathcal{D}_{\text{loc}}^e} d_e(z_e, z_i) \mid \forall e \in [1 \dots M] \right\}. \quad (34)$$

We refer to $z_{ie}^* = (\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*)$ as the material state associated to e , defined as the material data point closest to the current local mechanical state $z_e = (\mathbf{F}_e, \mathbf{P}_e)$:

$$d_e(z_e, z_{ie}^*) \leq d_e(z_e, z_i), \quad \forall e \in [1 \dots M], \forall z_i \in \mathcal{D}_{\text{loc}}^e. \quad (35)$$

The integer index ie is then the actual optimization variable. It maps the e^{th} integration point to the optimal i^{th} state pair in $\mathcal{D}_{\text{loc}}^e$.

- Let us now consider the stage where the material state $z^* = \{z_{ie}^* = (\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*)\}_{e=1}^M \in \mathcal{D}$ is determined. (31) can then be expressed as a *constrained* minimization problem. The solution set \mathcal{S}^{mec} of mechanical states is then given as

$$\mathcal{S}^{\text{mec}}(z^*) = \arg \min_{z \in \mathcal{Z}} \sum_{e=1}^M w_e d_e(z_e, z_{ie}^*) \quad (36a)$$

subject to

$$\mathbf{F}_e = \nabla_0^e \mathbf{u} + \mathbf{I}, \quad \forall e \in [1 \dots M], \quad (36b)$$

$$\mathbf{u}_a \cdot \mathbf{e}_i - u_\alpha^{\text{D}} = 0, \quad \forall (a, i) : \alpha(a, i) \in D, \quad (36c)$$

$$\sum_{e=1}^M w_e P_{ij}^e B_J^{ea} - \mathbf{T}_a^{\text{ext}} \cdot \mathbf{e}_i = 0, \quad \forall (a, i), \alpha(a, i) \notin D, \quad (36d)$$

$$\mathbf{r}_e = \boldsymbol{\epsilon} : (\mathbf{F}_e \mathbf{P}_e^T) = \mathbf{0}, \quad \forall e \in [1 \dots M], \quad (36e)$$

where, in (36d), the Einstein summation convention on repeated indices has been used.

As compared to the Lagrangian (small strain) formulation of [29] ([9]), the first minimization problem (34) is unchanged, except for the different local phase space, thanks to the quadratic deviation functions (16): the Green-Lagrange (linearized) strain tensor is merely replaced by the deformation gradient tensor, while the second Piola-Kirchhoff (Cauchy) stress tensor is substituted with the first Piola-Kirchhoff stress tensor. The second minimization problem (36) however differs from its Lagrangian counterpart in two ways: (i) the compatibility constraint (36b) is linear with respect to \mathbf{u} , and (ii) the conservation of angular momentum (36e) constitutes an additional constraint to the minimization problem, at integration points. The former feature is very advantageous but the latter one is difficult to implement.

On the one hand, the optimization of the material states is addressed with a tree-based nearest neighbor search algorithm, taking advantage of the quadratic deviation functions (16). As aforementioned, this approach drastically diminishes the computational cost of this part of the data-driven solver (DD solver). Indeed, the local phase space $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$ consisting of pairs of non-symmetric tensors in dimension n , it can be recast into the standard Euclidean space \mathbb{R}^{2n^2} , using the Mandel vector representation of the deformation gradient and stress tensors [32]. In that case, the search for nearest neighbor is to be performed M times (for every integration point) in a database of m samples of dimension $2n^2$. Using a brute-force search algorithm, this task grows as $O[Mm]$. With a tree-based data structure, the computational cost can be reduced to $O[M \log m]$ [33]. In practical DDCM applications, $m \gg M$ and the same material database is used for all integration points in the mesh (when a single material is considered); the tree-based data structure can be initialized once and for all at the beginning of the simulation. This approach thus provides a substantial gain in computation time. The interface that we used in [31] and herein, developed by [34], requires the data to be transformed so that the data-driven distance in local phase space

$\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$ is equivalent to the Euclidean distance in \mathbb{R}^{2n^2} . We refer to this transformation as the Euclidean mapping, which is detailed in Appendix B.

On the other hand, we could not address the constrained minimization of the mechanical state with the same approach as in [9, 29, 31]: the bi-linearity and local nature of the rotational equilibrium prevented us from using the method of Lagrange multipliers. Such an approach would add many unknowns to the numerical system as one would need an additional Lagrange multiplier vector attached to the integration points. Moreover, the injection of the stress correction equation as in [31] into the stationary equations would yield intractable residual vectors. As the addition of unknowns is not favorable, we turned to penalty methods. We elaborated an augmented Lagrangian approach, which offers more robustness than the quadratic penalty function method [35]. The details of the implementation are given next.

2.5. Augmented Lagrangian approach

The compatibility constraint (36b) is directly enforced by replacing the deformation gradient with its relationship to the displacement field. The unknowns in (36) then become the displacement field $\mathbf{u}(\mathbf{X})$ and the stress field $\mathbf{P}(\mathbf{X})$. Thanks to the FE discretization, the actual unknowns are the nodal displacements $\{\mathbf{u}_a\}_{a=1}^N$ and the values of the stress tensor at the integration points $\{\mathbf{P}_e\}_{e=1}^M$. These are recast into arrays u and P , of shape $Nn \times 1$ and $Mn^2 \times 1$ respectively, as

$$u = [u_{1,1}, u_{1,2}, u_{1,3}, \dots, u_{a,1}, u_{a,2}, u_{a,3}, \dots, u_{N,1}, u_{N,2}, u_{N,n}]^T \quad (37a)$$

$$P = [P_{11}^1, P_{12}^1, P_{13}^1, P_{21}^1, P_{22}^1, \dots, P_{33}^1, \dots, \quad (37b)$$

$$P_{11}^e, P_{12}^e, P_{13}^e, P_{21}^e, P_{22}^e, \dots, P_{33}^e, \dots, \quad (37c)$$

$$P_{11}^M, P_{12}^M, P_{13}^M, P_{21}^M, P_{22}^M, \dots, P_{33}^M]^T, \quad (37d)$$

where $u_{a,i}$ is the displacement of node a in direction i and P_{ij}^e is the (i, j) -th component of the stress tensor \mathbf{P}_e . The unknown array x is the concatenation of u and P

$$x = \begin{pmatrix} u \\ P \end{pmatrix}. \quad (38)$$

For a two-dimensional problem ($n = 2$), the array x is obviously reduced accordingly. In general, the total number of independent variables to determine in the minimization is then $(Nn) \times (Mn^2)$. The objective function of the minimization problem is denoted $f(u, P)$ and writes

$$f(u, P) = \sum_{e=1}^M w_e d_e (V(\mathbf{F}_e(u) - \mathbf{F}_{ie}^*) + V(\mathbf{P}_e - \mathbf{P}_{ie}^*)). \quad (39)$$

The remaining minimization constraints to enforce are (i) the Dirichlet conditions (36c), (ii) the translational equilibrium (36d) and (iii) the conservation of angular momentum (36e).

In like manner, we store these constraints in dedicated arrays, defined respectively as

$$h^D = Su - u_D \quad (40a)$$

$$h^{\text{teq}} = \bar{B}^T WP - \bar{T}^{\text{ext}} \quad (40b)$$

$$h^{\text{req}} = [r_{1,1}, \dots, r_{e,i}, \dots, r_{M,n_r}]^T, \quad (40c)$$

where S is a $n_D \times (Nn)$ selection matrix of 0 and 1 ($S_{k\alpha} = 1$ if $\alpha = \alpha_k \in D$ and 0 otherwise), the matrix B collects the vectors \mathbf{B}_{ea} and encodes the geometry and connectivity of the mesh, W is the diagonal matrix of weights w_e such that $WP = \sum_{e=1}^M w_e P_{ij}^e$, and T^{ext} is the array of nodal external applied forces, flattened in like manner as u . The bar symbol $\bar{\cdot}$ over B and T^{ext} means that they have been reduced to the degrees of freedom only. Finally, n_r is the dimension of the vector \mathbf{r}_e measuring the conservation of angular momentum (see (36e)). In short, $n_r = 1$ for a two-dimensional problem and $n_r = 3$ for a three-dimensional problem. Then, h^D is an array of shape $(n_D \times 1)$, h^{teq} is an array of shape $(n_{\text{dof}} \times 1)$ and h^{req} is an array of shape $((Mn_r) \times 1)$, with n_{dof} the total number of degrees of freedom (note that $n_{\text{dof}} + n_D = Nn$). With each type of constraints, we associate a dedicated penalty parameter and a Lagrange multiplier vector of appropriate shape, as summarized in Table 1.

Table 1: Penalty parameters and Lagrange multipliers associated with the optimization constraints of the nominal formulation.

Constraint	Penalty parameter	Lagrange multiplier
Dirichlet conditions $h^D = Su - u_D$	c^D	λ , shape $(n_D \times 1)$
Translational equilibrium $h^{\text{teq}} = \bar{B}^T WP - \bar{T}^{\text{ext}}$	c^{teq}	η , shape $(n_{\text{dof}} \times 1)$
Angular momentum $h^{\text{req}} = [[\mathbf{r}_e]_i]$	c^{req}	μ , shape $(Mn_r \times 1)$

The constrained minimization problem (36) is then conveniently re-written as

$$\text{minimize } f(u, P) \quad \text{subject to } h = \begin{pmatrix} h^D \\ h^{\text{teq}} \\ h^{\text{req}} \end{pmatrix} = 0. \quad (41)$$

The corresponding augmented Lagrangian function is

$$L_{(c^D, c^{\text{teq}}, c^{\text{req}})}(x, (\lambda, \eta, \mu)) = f(u, P) + \frac{1}{2}c^D |h^D(x)|^2 + \frac{1}{2}c^{\text{teq}} |h^{\text{teq}}(x)|^2 + \frac{1}{2}c^{\text{req}} |h^{\text{req}}(x)|^2 \\ - \lambda^T h^D(x) - \eta^T h^{\text{teq}}(x) - \mu^T h^{\text{req}}(x). \quad (42)$$

A typical iteration of the approach then writes³

$$\lambda_{j+1} = \lambda_j - c_j^D h^D(x_j), \quad c_{j+1}^D = \beta c_j^D \quad (43a)$$

$$\eta_{j+1} = \eta_j - c_j^{\text{teq}} h^{\text{teq}}(x_j), \quad c_{j+1}^{\text{teq}} = \beta c_j^{\text{teq}} \quad (43b)$$

$$\mu_{j+1} = \mu_j - c_j^{\text{req}} h^{\text{req}}(x_j), \quad c_{j+1}^{\text{req}} = \beta c_j^{\text{req}} \quad (43c)$$

with x_j the solution of the unconstrained minimization of $L_{(c_j^D, c_j^{\text{teq}}, c_j^{\text{req}})}(x, (\lambda_j, \eta_j, \mu_j))$. The iterations (43) stop when the Karush-Kuhn-Tucker (KKT) conditions are satisfied within a given tolerance:

$$\begin{pmatrix} \nabla_u f \\ \nabla_P f \end{pmatrix} (x) - (\nabla h(x))^T \begin{pmatrix} \lambda \\ \eta \\ \mu \end{pmatrix} < \text{tol}. \quad (44)$$

The unconstrained minimization of the augmented Lagrangian function (42) is performed with a Newton-Raphson (NR) scheme, considering that the gradient and Hessian matrix can be explicitly derived and rather easily implemented. A typical NR iteration writes:

$$x^{(j+1)} = x^{(j)} + \delta x^{(j)}, \quad (45a)$$

with

$$\begin{pmatrix} \nabla_u L \\ \nabla_P L \end{pmatrix} + \begin{pmatrix} \nabla_u^2 L & \nabla_P \nabla_u L \\ \nabla_u \nabla_P L & \nabla_P^2 L \end{pmatrix} (x^{(j)}) \delta x^{(j)} = 0, \quad (45b)$$

where the expressions of the gradient vectors $\nabla_u L$ and $\nabla_P L$ and Hessian matrices $\nabla_u^2 L$, $\nabla_P \nabla_u L$, $\nabla_u \nabla_P L$, and $\nabla_P^2 L$ are given in Appendix A. Note that this approach requires to build and solve a $(n \times N + n_{\text{states}}) \times (n \times N + n_{\text{states}})$ linear system of algebraic equations at every iteration of the NR scheme, every step of the augmented Lagrangian process, every iteration of the data-driven alternated minimization, where $n_{\text{states}} = n^2 \times M$ is the number of stress unknowns (n^2 independent components in the stress tensor of all M integration points).

2.6. Linear solver

A simpler version of the solver consists in enforcing the compatibility and translational equilibrium only, *i.e.* minimizing the mechanical states in \mathcal{E}_0 instead of \mathcal{E} . In that case, the augmented Lagrangian function is reduced to

$$L_{(c^D, c^{\text{teq}})}(x, (\lambda, \eta)) = f(u, P) + \frac{1}{2} c^D |h^D(x)|^2 + \frac{1}{2} c^{\text{teq}} |h^{\text{teq}}(x)|^2 - \lambda^T h^D(x) - \eta^T h^{\text{teq}}(x). \quad (46)$$

³To be consistent with the physical interpretation of the Lagrange multiplier $\boldsymbol{\eta}$ derived in the small strain or Lagrangian formulations, we chose to use “negative” Lagrange multipliers. The updating rule is modified in consequence, without loss of consistency with the original method of [35]. The nodal vectors $\boldsymbol{\eta}_a$ are then again interpreted as virtual displacements which, associated to a stiffness C , generate stresses that correct the out-of-balance gap between the material stresses and the external nodal forces.

The stationary equations yield two independent linear systems

$$\nabla_u L = \nabla_u f + c^D (\nabla_u h^D)^T h^D - (\nabla_u h^D)^T \lambda = 0, \quad (47a)$$

$$\nabla_P L = \nabla_P f + c^{\text{teq}} (\nabla_P h^{\text{teq}})^T h^{\text{teq}} - (\nabla_P h^{\text{teq}})^T \eta = 0. \quad (47b)$$

The pseudo-stiffness matrix of the two systems are respectively of size $Nn \times Nn$ and $n_{\text{states}} \times n_{\text{states}}$. This is computationally more advantageous than the complete nominal solver derived above. However, we need to assess whether the conservation of angular momentum is sufficiently enforced in the material data set \mathcal{D} rather than in the constraint set \mathcal{E} . Indeed, the constraint of rotational equilibrium $\mathbf{r}_e = \mathbf{0}$ is purely local. It could then be verified at each integration point in the mesh thanks to the associated material state instead of enforced in the minimization of the mechanical state.

2.7. Algorithms and summary

The complete augmented Lagrangian scheme is listed in Algorithm 1. It fits in stage 2 of the alternated minimization listed in Algorithm 2. Note that the DD solver Algorithm 2 is embedded in an incremental loading loop, not shown.

Along with its Lagrangian counterpart[31], the finite strain data-driven solver in the nominal formulation presents two main features:

- the computation of the mechanical states in two steps, namely
 - the resolution of the constrained minimization problem (41) with an augmented Lagrangian method inspired from [35], which uses a NR scheme for the minimization of the augmented Lagrangian function,
 - the computation of the local deformation gradients only, *via* the compatibility (25a), the stress being up-to-date as unknowns of the above minimization;
- the nearest-neighbor search for the optimal states in the material database from (35), using a tree-based search algorithm.

3. Numerical results

This section discusses some of the practical aspects in using finite strain data-driven solvers. To this end, we build artificial material databases from a constitutive relation and compare the data-driven results to classical finite element analysis achieved from the same model. It is then expected that the data-driven solution recovers or at least converges to the classical solution, which we take as a reference solution. To reduce the dimensionality of the phase space, we only consider two-dimensional problems, assuming plane stress conditions and incompressibility of the material. In the following, all tensors are then two-dimensional.

To generate the standard FE response, we use the standard neo-Hookean model. As shown in Appendix C, under plane stress and incompressibility constraints, the in-plane

Algorithm 1 Augmented Lagrangian solver for the nominal formulation of the data-driven constrained minimization problem.

INPUT: Results from the $(k - 1)$ -th iteration of the data-driven solver.

OUTPUT: Displacement and stress fields satisfying mechanical admissibility.

Require: Initial penalty parameters $c_D^0, c_0^{\text{teq}}, c_0^{\text{req}}$, initial Lagrange multipliers λ_0, η_0, μ_0 , tolerance parameter $r_{\text{tol}} > 0$, maximum number of iterations j_{max} .

```

1:  $j \leftarrow 0$ 
2:  $x_{j=0} \leftarrow u^{(k-1)}, P^{(k-1)}$  ▶ start from the results of the previous data-driven iteration
3:  $c_{j=0}^D, c_{j=0}^{\text{teq}}, c_{j=0}^{\text{req}} \leftarrow c_0^D, c_0^{\text{teq}}, c_0^{\text{req}}$ 
4:  $\lambda_{j=0}, \eta_{j=0}, \mu_{j=0} \leftarrow \lambda_0, \eta_0, \mu_0$ 
5:  $\text{KKT}_{j=0} \leftarrow \|\nabla f(x_0) - [\lambda_0^T, \eta_0^T, \mu_0^T] \nabla h(x_0)\|_2$  ▶ initialize KKT conditions
6: for  $j = 1 \dots j_{\text{max}}$  do ▶ Augmented Lagrangian iterations
7:    $x_j \leftarrow \text{minimize } L_{c_j^D, c_j^{\text{teq}}, c_j^{\text{req}}}(x, (\lambda_j, \eta_j, \mu_j))$  ▶ NR (45) with initial guess  $x_{j-1}$ 
8:    $\text{KKT}_j \leftarrow \|\nabla f(x_j) - [\lambda_j^T, \eta_j^T, \mu_j^T] \nabla h(x_j)\|_2$  ▶ update KKT conditions
9:   if  $\text{KKT}_j \leq r_{\text{tol}} \text{KKT}_0$  or  $\text{KKT}_j > \text{KKT}_{j-1}$  then
10:     return  $x_j$ 
11:   else
12:     update Lagrange multipliers
13:      $\lambda_{j+1} \leftarrow \lambda_j - c_j^D h^D(x_j)$ 
14:      $\eta_{j+1} \leftarrow \eta_j - c_j^{\text{teq}} h^{\text{teq}}(x_j)$ 
15:      $\mu_{j+1} \leftarrow \mu_j - c_j^{\text{req}} h^{\text{req}}(x_j)$ 
16:     update penalty parameters [35]
17:     if  $\|h(x_j)\|_2 > \gamma \|h(x_{j-1})\|_2$  then
18:        $c_{j+1} \leftarrow \beta c_j$ 
19:     else
20:        $c_{j+1} \leftarrow c_j$ 
21:     end if
22:   end if
23: end for

```

Algorithm 2 Data-driven solver - nominal formulation.

INPUT: A mesh of M integration points and N nodes, boundary and loading conditions, material data set \mathcal{D} .

OUTPUT: mechanical response (displacements $\{\mathbf{u}_a\}_{a=1}^N$, mechanical states $\{(\mathbf{F}_e, \mathbf{P}_e)\}_{e=1}^M$), material states $\{(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*)\}_{e=1}^M$.

Require: maximum number of iterations k_{\max} .

1) *Initialize all material states and mapping*

1: $k \leftarrow 0$

2: **for** $e = 1 \dots M$ **do**

3: $ie^{(0)} \leftarrow i, i \in [1 \dots m_e]$

4: **end for**

5: $u^{(k=0)} \leftarrow 0$ and $P^{(k=0)} \leftarrow 0$

6: **for** $k = 1 \dots k_{\max}$ **do**

 2) *Compute mechanical states*

 2a) *Minimize (41)*

7: $(u^{(k)}, P^{(k)}) \leftarrow x$ from Algorithm 1

 2b) *Update mechanical states*

8: **for** $e = 1 \dots M$ **do**

9: $\mathbf{P}_e^{(k)} \leftarrow P^{(k)}$

10: $\mathbf{F}_e^{(k)} \leftarrow \nabla_0^e \mathbf{u}^{(k)} + \mathbf{I}$

11: **end for**

 3) *Update material states and mapping*

12: **for** $e = 1 \dots M$ **do**

13: $ie^{(k)} \leftarrow (\mathbf{F}_{ie^{(k)}}^*, \mathbf{P}_{ie^{(k)}}^*)$ from (35)

14: **end for**

 4) *Test convergence*

15: **if** $ie^{(k)} = ie^{(k-1)}$ for all $e = 1 \dots M$ **then**

16: $\mathbf{u} \leftarrow \mathbf{u}^{(k)}$

17: $(\mathbf{F}_e, \mathbf{P}_e) \leftarrow (\mathbf{F}_e^{(k)}, \mathbf{P}_e^{(k)})$ for all $e = 1 \dots M$

18: **exit.**

19: **else**

20: $ie^{(k+1)} \leftarrow ie^{(k)}$ for all $e = 1 \dots M$

21: **end if**

22: **end for**

second Piola-Kirchhoff stress tensor $\mathbf{S} \in \mathbb{R}_{\text{sym}}^{2 \times 2}$ is expressed as a function of the in-plane right Cauchy-Green stretch tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F} \in \mathbb{R}_{\text{sym}}^{2 \times 2}$ by

$$\mathbf{S} = \mu (\mathbf{I} - (\det \mathbf{C})^{-1} \mathbf{C}^{-1}), \quad (48)$$

where $\mu = 1.2 \text{ MPa}$ is the shear modulus.

We investigate three 2D test cases: (i) uniaxial tension of a thin membrane, (ii) uniaxial tension of a thin membrane with the bottom clamped, and (iii) more complex loading (simultaneous tension and shear) of a thin membrane with a hole.

3.1. Validation: uniaxial tension

The first test case is used to validate the nominal DD solver in the simplest solicitation possible: a uniaxial tensile test of a thin hyperelastic membrane. It also enables the visualization of the phase space in two dimensions and an easy synthetic generation of the database. Indeed, the stress tensors only have one non-zero component and the deformation-gradient tensors are entirely characterized by two inter-dependent quantities (longitudinal and transversal stretch ratios).

3.1.1. Computational problem

We consider a rectangular thin membrane of dimensions $l \times h = 5 \times 10 \text{ mm}^2$, under plane stress conditions. The problem is then reduced to two dimensions. The geometry and mesh of the sample are shown in Fig. 1. The mesh comprises $N = 36$ nodes and 23

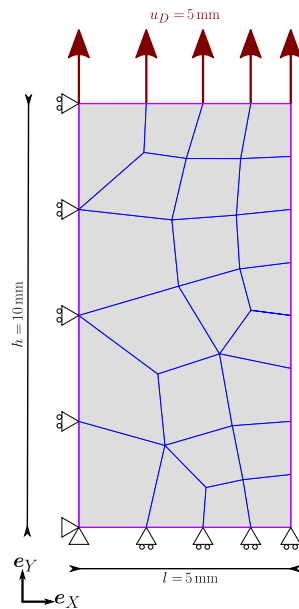


Figure 1: Geometry, boundary conditions and mesh for the uniaxial tensile test.

bi-linear quadrangular elements with 4 integration points each, which amounts to $M = 92$

integration points. The boundary conditions are also represented in Fig. 1 and write

$$\mathbf{u}_a \cdot \mathbf{e}_Y = 0, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, 0) \quad (49a)$$

$$\mathbf{u}_a \cdot \mathbf{e}_X = 0, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (0, Y_a) \quad (49b)$$

$$\mathbf{u}_a \cdot \mathbf{e}_Y = u_D, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, h) \quad (49c)$$

with $u_D = 5$ mm which corresponds to a stretch $\lambda_Y^{\text{ref}} = 1.5$.

The material database is generated by a sampling of the standard uniaxial stretch-stress curve of the neo-Hookean model. From (48) and the relation $\mathbf{P} = \mathbf{F}\mathbf{S}$, it writes

$$\mathbf{P}(\lambda_Y) = \mu (\lambda_Y - \lambda_Y^{-2}) \mathbf{e}_Y \otimes \mathbf{e}_Y, \quad (50)$$

with

$$\mathbf{F}(\lambda_Y) = \lambda_X \mathbf{e}_X \otimes \mathbf{e}_X + \lambda_Y \mathbf{e}_Y \otimes \mathbf{e}_Y. \quad (51)$$

where λ_Y and $\lambda_X = 1/\sqrt{\lambda_Y}$ are the longitudinal and transversal stretch ratios respectively. All integration points in the mesh share the same local material data set \mathcal{D}_{loc} , thus defined as

$$\mathcal{D}_{\text{loc}} = \{(\mathbf{F}, \mathbf{P}) \mid \mathbf{F} = \mathbf{F}(\lambda_Y) \text{ from (51), } \mathbf{P} = \mathbf{P}(\lambda_Y) \text{ from (50)}\}. \quad (52)$$

The material database $\mathcal{D}_{\text{loc}}^m$ results from a sampling of m points in \mathcal{D}_{loc} , *i.e.* a sampling of m longitudinal stretch ratios λ_Y , evenly spaced in the interval $[0.9, 2]$. We generate two types of databases:

1. A first database $\mathcal{D}_{\text{loc}}^{100}$, which contains the reference solution at $\lambda_Y^{\text{ref}} = 1.5$ together with 99 other material data points.
2. A second family of database, which does not contain the reference solution and consists of several samplings of increasing density: $m \in \{11, 51, 101, 1001, 10\,001, 100\,001\}$.

Finally, the parameters of the DD solver are as follows:

- The amplitude of the scaling parameter C is set to the linearized Young's modulus of the model: $C = 3\mu$.
- The material states are initialized to zero deformation state (see Line 3 in Algorithm 2): $(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*) = (\mathbf{I}, \mathbf{0})$, $\forall e \in [1 \dots M]$,
- The initial penalty parameters c_0^D, c_0^t, c_0^m are respectively set to $1 \times 10^3 C$, $1 \times 10^3 C$, C . The initial Lagrange multipliers λ_0, η_0, μ_0 are all set to zero (see Lines 3 and 4 in Algorithm 1).

3.1.2. Results

First, we demonstrate in Fig. 2 that the reference solution is recovered for the material database $\mathcal{D}_{\text{loc}}^{100}$, which contains the reference solution. The DD solver converged within only $k = 5$ iterations of the alternated minimization. In addition, the mechanical states (rev circles) always remain on the constraint set throughout the iterations, while the material states are successively selected as the closest data points. This is illustrated by the projection

$P_{\mathcal{D}}$ and $P_{\mathcal{E}}$ on the material data set and on the constraint set (dotted and solid gray lines) respectively. On the one hand, the slope of the projection $P_{\mathcal{D}}$ is controlled by the C -parameter. On the other hand, the slope of the projection $P_{\mathcal{E}}$ reveals that the computation of the mechanical states for this displacement-controlled problem boils down to a correction of the stretch ratio λ_Y . In Fig. 2, we also illustrate the deformation of the mesh throughout the iterations (purple lines). We can see that, in that case, the first iteration consists in enforcing the boundary conditions (semi-transparent purple lines at the furthest right delimits the deformed mesh at the end of the first iteration $k = 1$).

After this sanity check with $\mathcal{D}_{\text{loc}}^{100}$, we proceed to show the results obtained from material data sets which do not contain the reference solution. In Fig. 3, we plot the root-mean-square (RMS) percent errors in strain and stress against the size of the material database, re-defined as

$$F_{(\% \text{RMS})} = \left(\frac{\sum_{e=1}^M w_e V (\mathbf{F}_e - \mathbf{F}_e^{\text{ref}})}{\sum_{e=1}^M w_e V (\mathbf{F}_e^{\text{ref}})} \right)^{\frac{1}{2}} \quad (53a)$$

$$P_{(\% \text{RMS})} = \left(\frac{\sum_{e=1}^M w_e V^* (\mathbf{P}_e - \mathbf{P}_e^{\text{ref}})}{\sum_{e=1}^M w_e V^* (\mathbf{P}_e^{\text{ref}})} \right)^{\frac{1}{2}}, \quad (53b)$$

with V and V^* as in (16). As expected, both errors converge to zero with increasing number of material data points⁴.

This very simple example has already shown interesting features of the finite strain data-driven solver:

- when the reference solution is in the data set, the DD solver is able to recover it;
- when the intersection between the constraint set and the material data set is empty, the data-driven solution improves with the number of material data points, as the best material data point gets increasingly closer to the reference solution.

3.2. Non homogeneous case: clamped tension

Let us now consider a slightly more complex problem. We perform data-driven simulations of a thin membrane in tension, which is clamped at the bottom. The mechanical fields should then be homogeneous in most of the structure, except near the clamped edge.

3.2.1. Computational problem

The problem is again two dimensional, as we make the assumption of plane stress conditions. The geometry, mesh and loading of the sample are given in Fig. 4. The mesh comprises $N = 340$ nodes and 378 bi-linear quadrangular elements with 4 integration points

⁴Note that, in the pre-asymptotic regime, the simulations can lead to the same solution depending on the value of C , as can be seen from the results corresponding to $\mathcal{D}_{\text{loc}}^{51}$ and $\mathcal{D}_{\text{loc}}^{101}$ in Fig. 3. Such numerical artifacts are not relevant to the convergence of the method and were thus ignored to fit the convergence slope (solid line in Fig. 3).

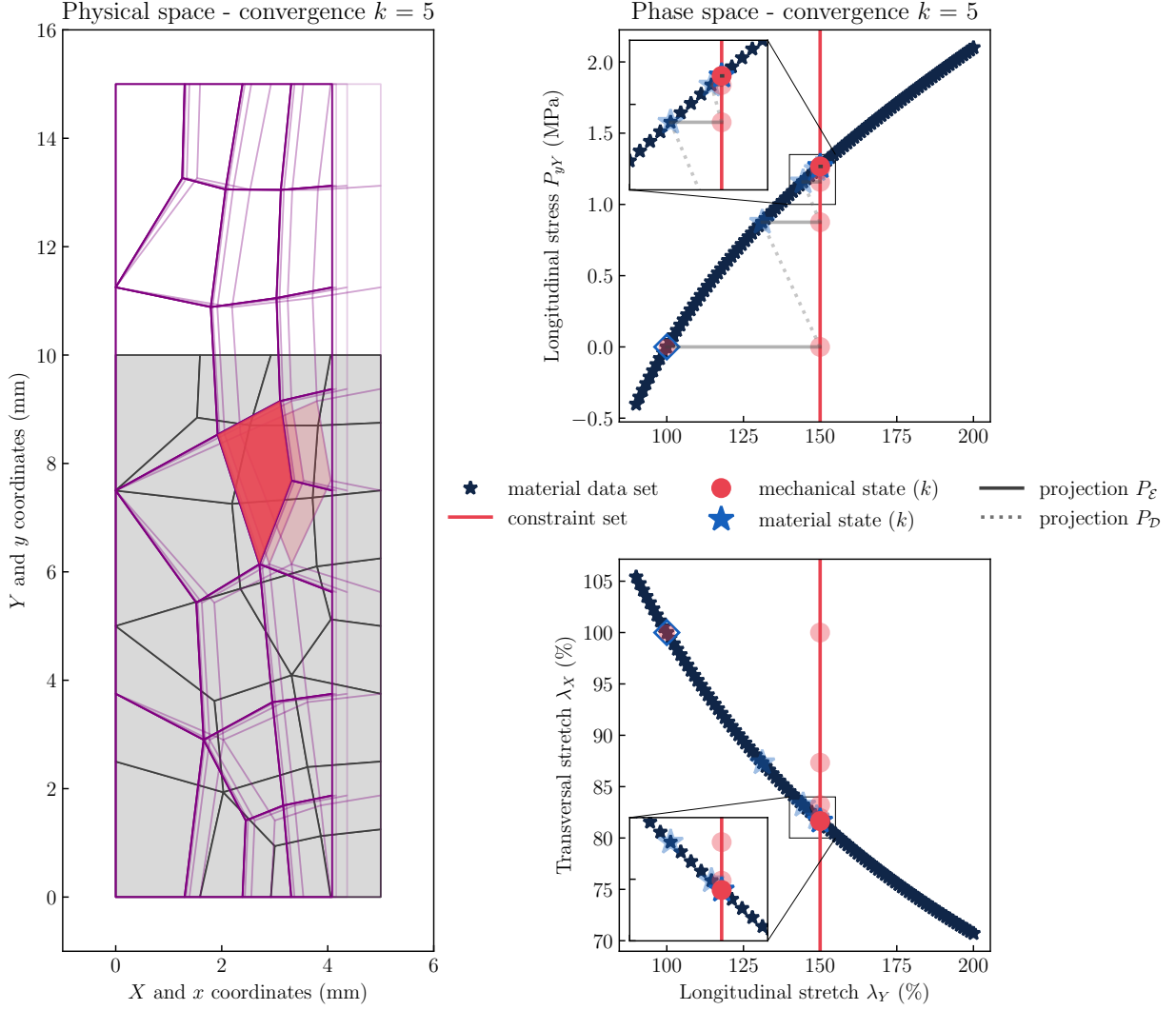


Figure 2: Physical space *vs* phase space representation of the data-driven results. Uniaxial tensile test, obtained for the data set $\mathcal{D}_{\text{loc}}^{100}$, which contains the reference solution. In phase space, we only plot the mechanical and material state of the corresponding rev colored element in physical space. The results from previous iterations are semi-transparent. The blue diamond indicates the initial material state. Note that, here, the represented constraint set is a projection of it in the (λ_Y, P_{yY}) plane. It is actually much more complex.

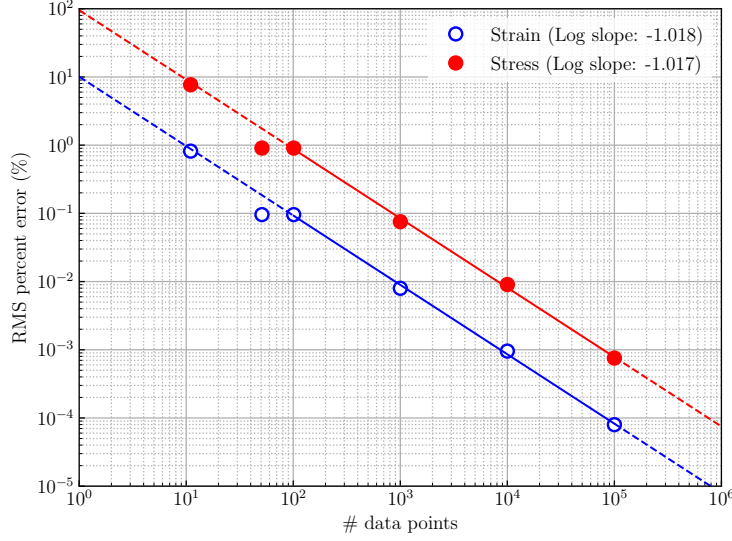


Figure 3: Convergence of the RMS percent errors in strain and stress with respect to the size of the database, for the uniaxial tensile test.

each, which amounts to $M = 1512$ integration points. The boundary conditions are also represented in Fig. 4 and write

$$\mathbf{u}_a = \mathbf{0}, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, 0); \quad (54)$$

$$\mathbf{u}_a \cdot \mathbf{e}_Y = u_D, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, h). \quad (55)$$

with $u_D = 5$ mm. In that case, the simulation is performed incrementally: the displacements are gradually prescribed in 10 regular steps.

The material database is only constituted with the reference FE solution of the problem, obtained with the incompressible neoHookean model (48). For all 10 load steps, we concatenate the reference deformation gradient-stress state of all integration points into the local material data set as

$$\mathcal{D}_{\text{loc}} = \left\{ \left\{ (\mathbf{F}_e^{\text{ref},t}, \mathbf{P}_e^{\text{ref},t}) \right\}_{e=1}^M \right\}_{t=1}^{10} \quad (56)$$

Note that the database then comprises 15 120 material data points.

The DD solution is achieved with the following parameters:

- Different initialization methods for the material states are evaluated:

“**zero**” all material states are initialized to the zero-deformation state:

$$(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*) = (\mathbf{I}, \mathbf{0}), \quad \forall e.$$

“**random**” all material states are initialized to the same random data point:

$$(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*) = (\mathbf{F}'_{i_r}, \mathbf{P}'_{i_r}), \quad \forall e, \quad \text{with } i_r \text{ a random integer in } [1 \dots m].$$

“**random point-wise**” each material state is initialized to a different random data point: $(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*) = (\mathbf{F}'_{e_r}, \mathbf{P}'_{e_r}), \quad \forall e$, with e_r a different random integer in $[1 \dots m]$ for each e .

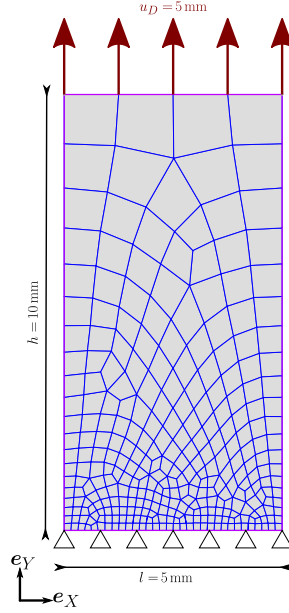


Figure 4: Geometry, boundary conditions and mesh for the clamped tensile test.

“reference” each material state is initialized to the corresponding reference solution:

$$(\mathbf{F}_{ie}^*, \mathbf{P}_{ie}^*) = (\mathbf{F}_e^{\text{ref}}, \mathbf{P}_e^{\text{ref}}), \forall e.$$

Note that this initialization holds for the first loading step only; afterwards, the initial material state is set to the converged material state of the previous loading step:

$$\left(\mathbf{F}_{ie}^{*,(0)}, \mathbf{P}_{ie}^{*,(0)}\right)^{(t)} = \left(\mathbf{F}_{ie}^{*,(k_{\max})}, \mathbf{P}_{ie}^{*,(k_{\max})}\right)^{(t-1)}.$$

- The initial penalty parameters c_0^D , c_0^t , c_0^m are respectively set to $1 \times 10^3 C$, $1 \times 10^3 C$, C . The initial Lagrange multipliers λ_0 , η_0 , μ_0 are all set to zero, as previously.
- All simulations are performed on a laptop with the following specifications: processor Intel Core i5-6200U CPU @ 2.30 GHz \times 4, with 15.5 Gio of RAM.

With these settings, two kinds of data-driven simulations are performed: (i) using the nominal solver, (ii) using the so-called “linear” solver, where the conservation of angular momentum is not enforced (see Section 2.6). The latter test is aimed at assessing whether this condition must be enforced in the constraint set when it is already satisfied by the material data set (see [30, Theorem 3.5]).

3.2.2. Results and discussion

We first consider the results obtained with the “zero” initialization method. Contrary to the previous example, the reference solution is not recovered in that case: the value of the objective function at convergence is not a numerical zero, as listed in Table 2. However, the solution is satisfactory enough, as shown in Fig. 5, by the maps of the Von Mises norm

Table 2: Value of the objective function at convergence, for the last load step of the clamped tensile test. The iterations correspond to the total number of local data assignment iterations throughout all 10 load steps.

Formulation	Objective function	Iterations	Computation time
Nominal	$1.74 \times 10^{-3} \text{ MPa mm}^2$	256	94 min
“Linear” nominal	$3.75 \times 10^{-3} \text{ MPa mm}^2$	175	19 min

of the Cauchy stress tensors (a good indicator of both strain and stress prediction since $\boldsymbol{\sigma} = J^{-1} \mathbf{P} \mathbf{F}^T$).

When comparing all initialization methods in Fig. 6, we demonstrate that the alternated minimization can diverge if the initial guess is too far from the solution. On the left-hand bottom corner of Fig. 6, we can see that, as expected, the “reference” initialization converged within one iteration and provides the global minimum of the objective function, which is a numerical zero. On the upper part of Fig. 6, we show the convergence of 8 and 10 different “random” and “random point-wise” initializations respectively. The initial value of the objective function (at $k = 1$) in these cases is already very high (about $1 \times 10^5 \text{ MPa mm}^2$) and most of the simulations diverged (as indicated by a cross at the end of a line)⁵. The most robust and accurate choice is “zero”, as can be seen from the center part of Fig. 6: the objective function smoothly decays from $2.24 \times 10^{-1} \text{ MPa mm}^2$ to $2.00 \times 10^{-4} \text{ MPa mm}^2$ within $k = 19$ iterations. The advantage of the “zero” method was expected, as we also set the initial guess for the displacements u and mechanical stress P to zero (see Line 5 in Algorithm 2). This choice was motivated by the fact that the DD solver is embedded in an incremental loading loop, as standard FE nonlinear solvers. In addition, we can show that such an initialization method corresponds to a linear elastic first estimate of the mechanical states, derived from the boundary and loading conditions only. Conversely, non-zero material states act as pre-strain and pre-stress conditions on the structure. Their values should then be consistent with the external loading. In an incremental loading procedure, it thus makes sense to set them to zero, as the first loading conditions will be low. This analysis only aims at showing that the “zero” initialization method is a safer practice than the “random” initialization suggested in previous studies, when using the alternated minimization. Indeed, the database used herein is very sparse and the alternated minimization is crude; with the development of more robust DD solvers and databases of infinite size and richness, the initial guess should not impact the results in the future.

We now evaluate the need to enforce the conservation of angular momentum in the constraint set \mathcal{E} . We refer to the vector $\mathbf{r} = \boldsymbol{\epsilon} : (\mathbf{F} \mathbf{P}^T)$ in (9) as the *rotational residual*. In addition, we define the *rotational residual percent error* $\epsilon_{(\% \text{rot})}$ as the ratio between the

⁵In fact, the initial material state is so far from the solution that the Newton-Raphson diverges right away. However, we let the simulation go on a bit to see if the alternated minimization can recover the path towards the solution, before stopping the simulation. This explains why the value of the objective function is increasing in some cases. Note that this issue is not possible in small strain as the minimization problem is quadratic with linear constraints.

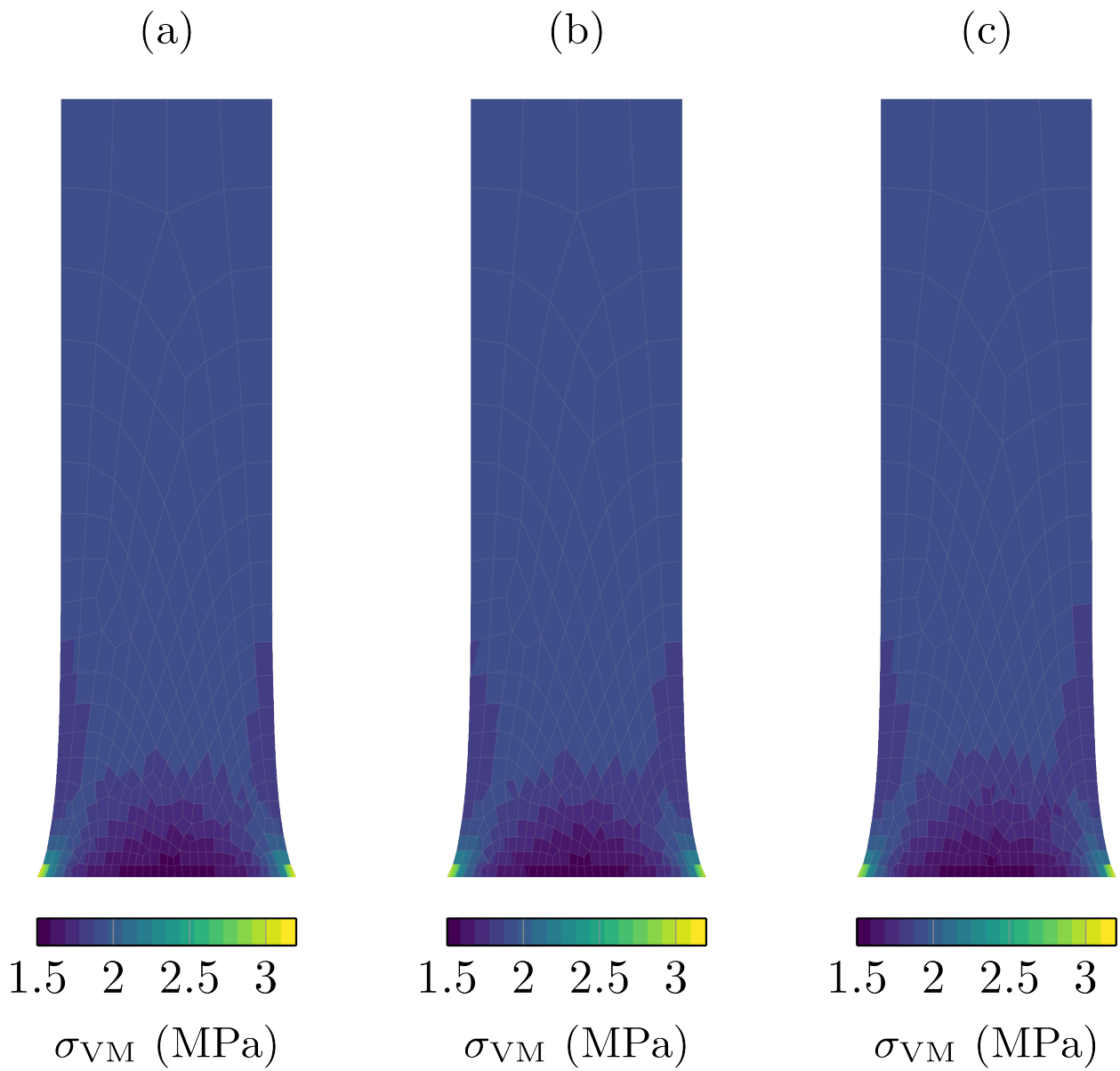


Figure 5: Von Mises norm of the Cauchy stress tensor, plotted on the deformed configuration. Only 18 filled-isovalues are shown. (a) FE reference solution. (b) Nominal solver solution. (c) “Linear” solver solution.

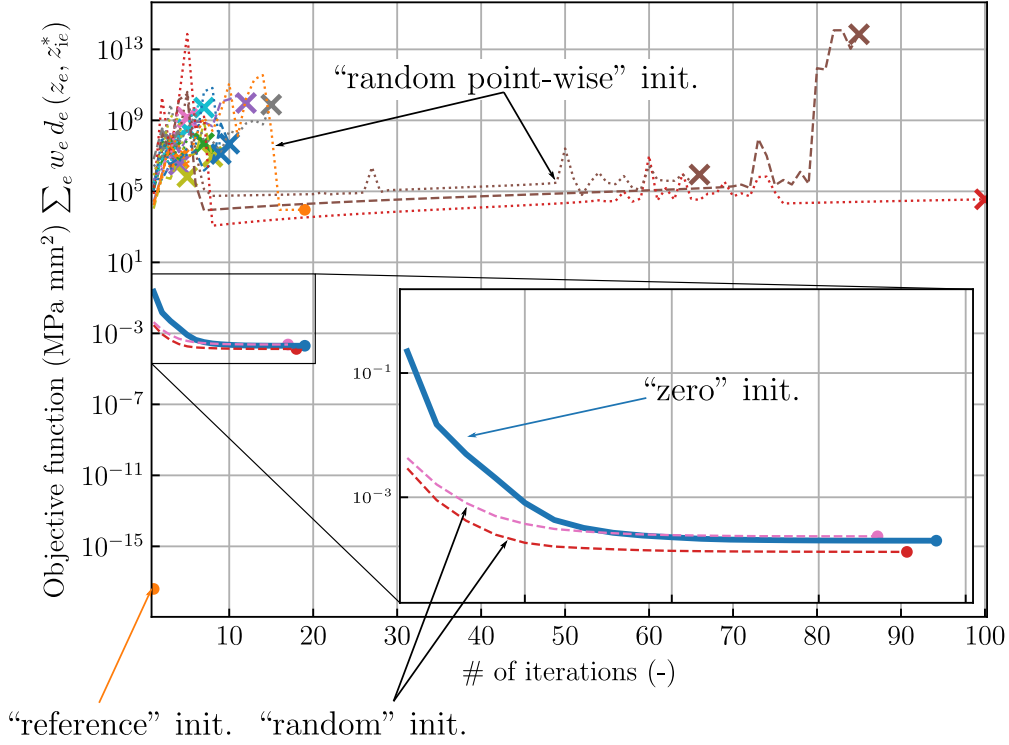


Figure 6: Results are given for the first load step only. Convergence of the data-driven finite strain nominal solver for the clamped tensile test. Comparison of different initialization methods: “zero” initialization is in solid line, “random” initializations are in dashed lines and “random point-wise” in dotted lines. A dot (respectively a cross) at the end of a line indicates that the solver did (respectively did not) converge.

rotational residual and a reference strain energy density:

$$\epsilon_{(\%rot)} = \frac{\|\boldsymbol{\epsilon} : (\mathbf{F}\mathbf{P}^T)\|_2}{\mathbf{F} : \mathbf{P}}. \quad (57)$$

Note that, in two-dimensional problems, the rotational residual is a one-dimensional vector, *i.e.* a scalar; specifically,

$$\|\boldsymbol{\epsilon} : (\mathbf{F}\mathbf{P}^T)\|_2 = |F_{11}^e P_{21}^e + F_{12}^e P_{22}^e - (F_{21}^e P_{11}^e + F_{22}^e P_{12}^e)|. \quad (58)$$

The rotational residual percent error is computed for both DD solutions, obtained from the nominal and “linear” solvers respectively. The results are given in Fig. 7, together with the rotational residual of the “linear” solver solution. As can be seen from Fig. 7(b), the “linear” solver does not recover the conservation of angular momentum. However, the error does not exceed 10% of the stored strain energy density $\mathbf{F} : \mathbf{P}$. Let recall that in the present case, the material database satisfies moment equilibrium since it was obtained from standard FE computations.

3.3. More complex case: membrane with a hole

In the previous example, the strain and stress fields were homogeneous in most of the structure, which could explain the relatively low impact of ignoring the rotational equilibrium

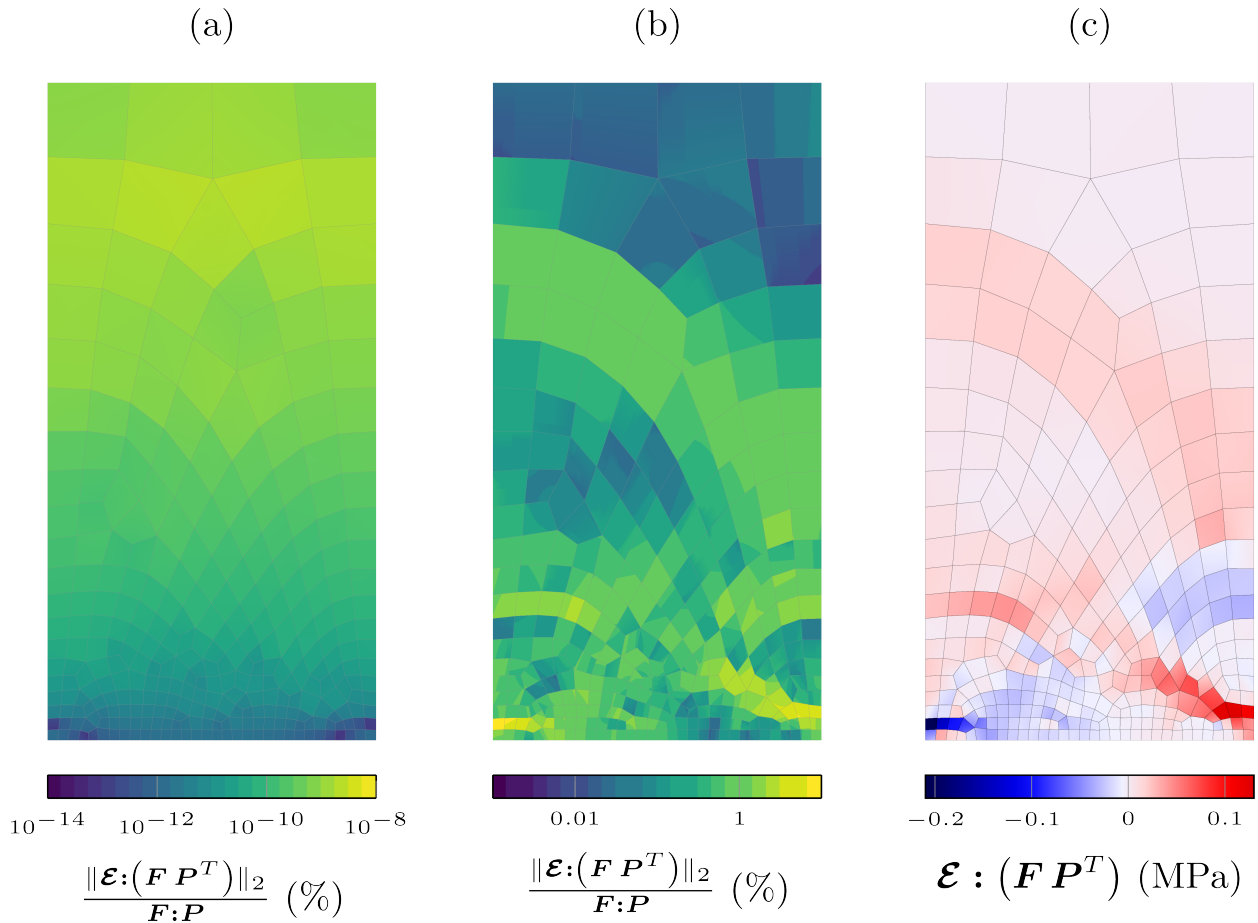


Figure 7: Conservation of angular momentum in the data-driven results for the clamped tensile test. (a) and (b) Rotational residual percent error. (a) Nominal solver solution. (b) “Linear” solver solution. The color map is in log scale. (c) Rotation residual (one-dimensional vector plotted as a scalar) for the “linear” solver.

constraint. We thus evaluate the solver with a more heterogeneous case.

3.3.1. Computational problem

The geometry, mesh and loading of the sample are given in Fig. 8. The mesh comprises $N = 1092$ nodes and 1015 bi-linear quadrangular elements with 4 integration points each, which amounts to $M = 4060$ integration points. The boundary conditions are also represented in Fig. 8 and write

$$\mathbf{u}_a = \mathbf{0}, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, 0); \quad (59)$$

$$\mathbf{u}_a = u_X^D \mathbf{e}_X + u_Y^D \mathbf{e}_Y, \quad \forall a \in [1 \dots N] : \mathbf{X}_a = (X_a, h); \quad (60)$$

with $u_X^D = 30$ mm and $u_Y^D = 5$ mm. The simulation is again performed incrementally: the displacements are gradually prescribed in 40 regular steps.

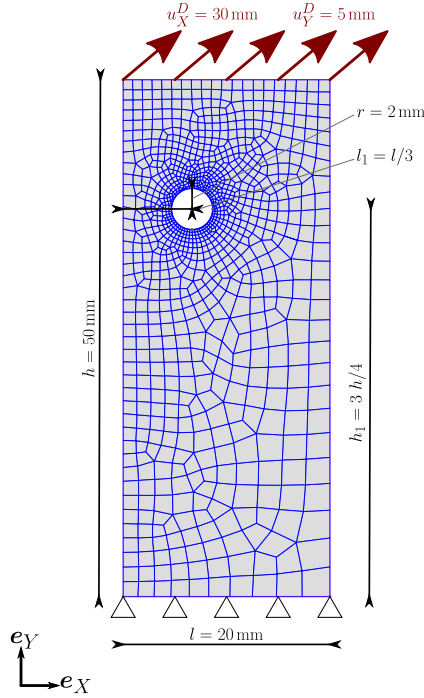


Figure 8: Geometry, boundary conditions and mesh for the membrane with a hole.

As previously, the material database is only constituted with the FE reference solution of the problem, obtained with the incompressible neoHookean model (48). For all 40 load steps, we concatenate the reference deformation gradient-stress state of all integration points into the local material data set as

$$\mathcal{D}_{\text{loc}} = \left\{ \left\{ (\mathbf{F}_e^{\text{ref},t}, \mathbf{P}_e^{\text{ref},t}) \right\}_{e=1}^M \right\}_{t=1}^{40} \quad (61)$$

Note that the database then comprises 162 400 material data points.

Based on the preceding analysis, the solution is achieved from the “zero” initialization method only. All other parameters of the simulations are unchanged from the previous example.

3.3.2. Results and discussion

Again the global minimum of the problem is not found by the alternated minimization, as shown by the values of the objective function listed in Table 3. Note that the computation time was divided by a factor 2 with the “linear” solver.

We again measure the rotational residual percent error in Fig. 9. In this more complex case, the error made by the “linear” solver is higher: it is mostly comprised between 1 % and 10 % but it can reach over 100 % around the holes.

The local error at integration points also has an impact on the global response of the structure, as measured by the loading curve of the simulation. We compare in Fig. 10 the displacement-force response of the clamped tensile test with the one of the complex

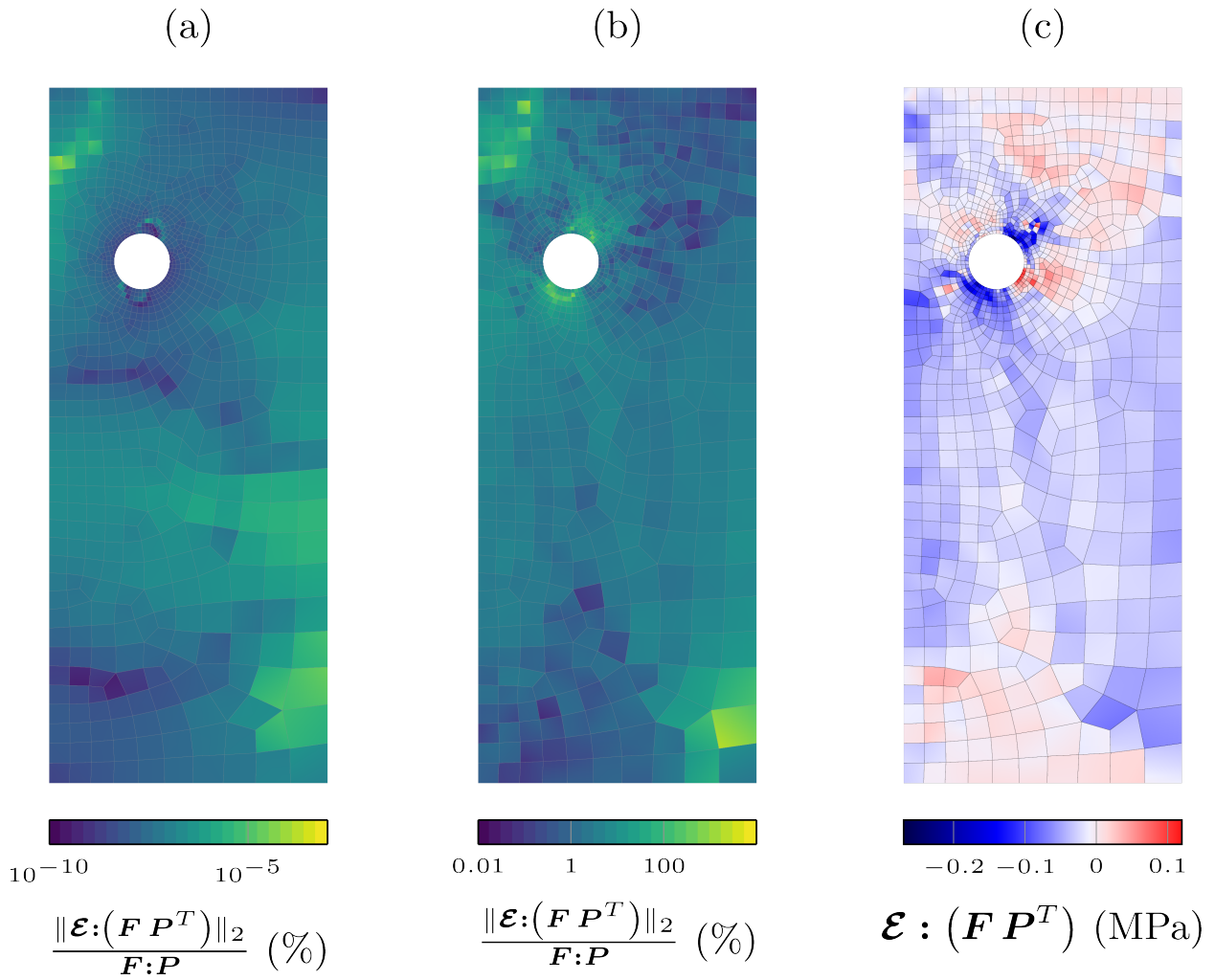


Figure 9: Conservation of angular momentum in the data-driven results for the membrane with a hole. (a) and (b) Rotational residual percent error. (a) Nominal solver solution. (b) “Linear” solver solution. The color map is in log scale. (c) Rotation residual (one-dimensional vector plotted as a scalar) for the “linear” solver.

Table 3: Value of the objective function at convergence, for the last load step of the complex loading case. The iterations correspond to the total number of local data assignment iterations throughout all of the 40 load steps.

Formulation	Objective function	Iterations	Computation time
Nominal	$2.46 \times 10^{-1} \text{ MPa mm}^2$	962	20.22 h
“Linear” nominal	$3.98 \times 10^{-1} \text{ MPa mm}^2$	1191	12.1 h

case for both solvers. In the clamped tensile test where most of the structure is subject

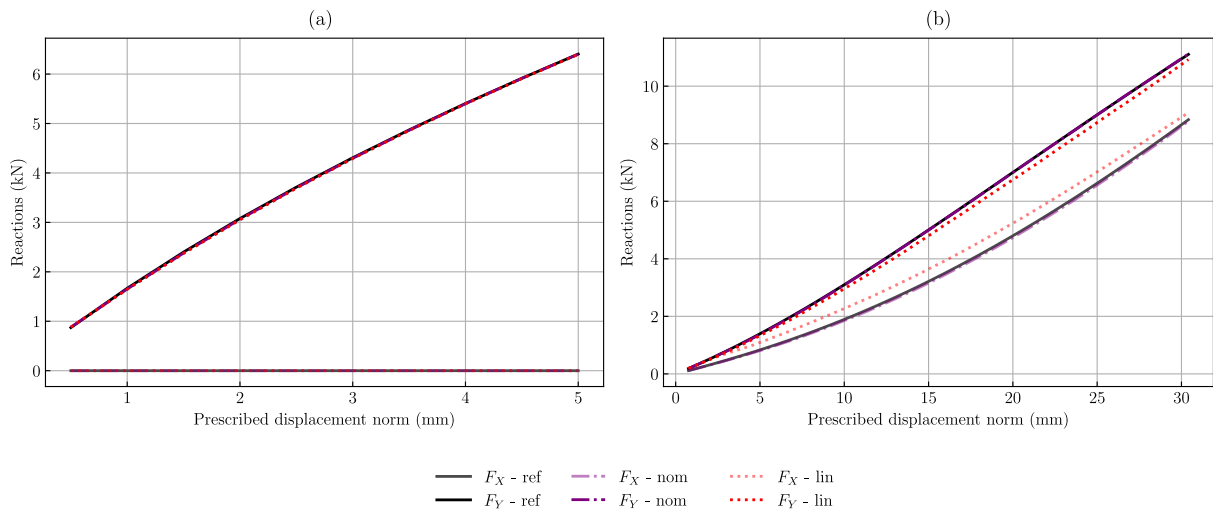


Figure 10: Loading curves for the clamped tensile test (a) and the complex case (b). In the legend, “nom” designates the nominal solver, “lin” designates the “linear” nominal solver and “ref” corresponds to the FE reference solution.

to uniaxial tension only, the impact of the error in rotational equilibrium on the reaction forces is invisible. Conversely, the local error made in the membrane with a hole weakens the accuracy of the reaction forces at the grip.

Finally, we recall that the conservation of angular momentum is, in theory, equivalent to the principle of material-frame indifference. Hence, it could be enforced in the material data set, rather than in the constraint set, by enriching the database with the orbits of every material data points. That is, the local material data set is now:

$$\mathcal{D}_{\text{loc}} = \left\{ \left\{ (QF_e^{\text{ref}}, QP_e^{\text{ref}}) \right\}_{e=1}^M \right\}_{t=1}^{40}, \quad (62)$$

with $Q \in SO(2)$. A discretization of $SO(2)$ is easily parameterized by one angle $\theta \in [0, 180)$. We compare the results obtained with three databases of increasing fidelity to material frame-indifference, generated *via* regular samplings of θ of decreasing step $\Delta\theta \in \{15^\circ; 10^\circ; 5^\circ\}$. In particular, we analyze in Fig. 11 the statistics of the rotational residual for each database.

As can be seen from the figure, the results are not improved with this method (in particular, median and mean values are almost unchanged).

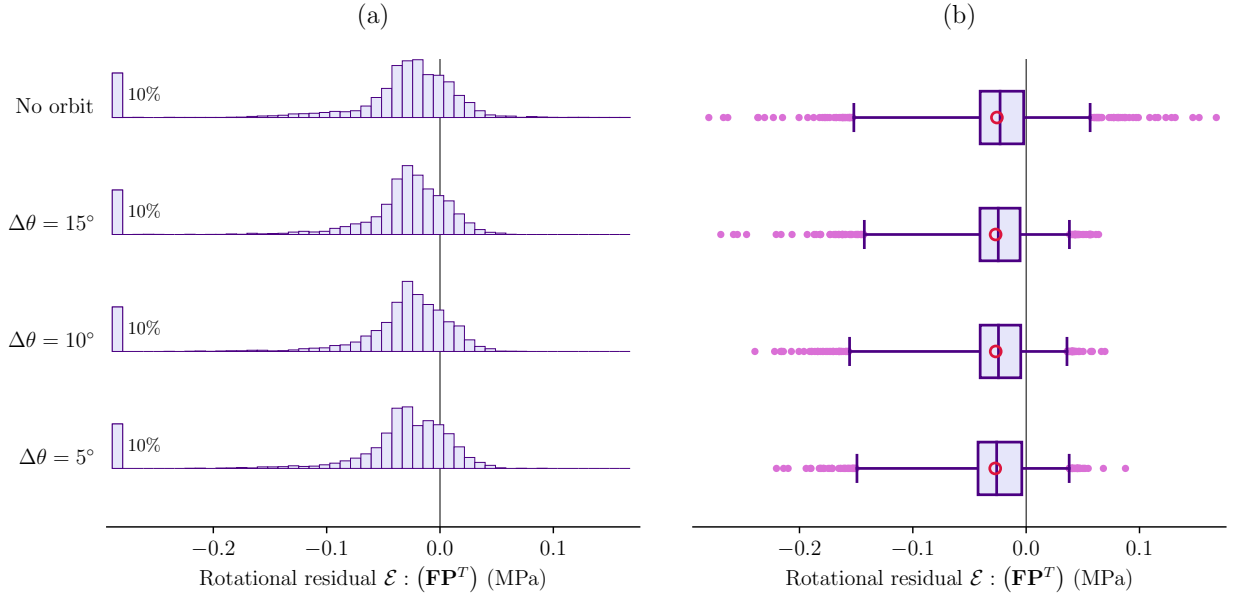


Figure 11: Rotational residual at every integration point. The databases are obtained with different angular discretizations $\Delta\theta$ of the orbit $Q \in SO(2)$. (a) Histogram of the data, in percent of number of integration points. At the furthest left, a bar corresponding to 10% of occurrence is given for scale. (b) The box extends from the 25th and 75th percentile of the data, with a line at the median. The red circle stands for the mean value. The whiskers show the 1th and 99th percentiles. Small purple dots represent data points out of that range (fliers).

4. Final remarks

The data-driven computing paradigm of [9] has been extended to finite strain elasticity by means of two competing formulations: (i) the so-called *Lagrangian* approach of [29] consists in minimizing a quadratic distance between Green-Lagrange strain–second Piola-Kirchhoff stress tensor pairs (\mathbf{E}, \mathbf{S}) in phase space, subject to nonlinear mechanical constraints; (ii) the so-called *nominal* definition of [30] minimizes a more general (p, q) –distance between deformation gradient–first Piola-Kirchhoff stress tensor pairs (\mathbf{F}, \mathbf{P}) , under linear constraints of compatibility and translational equilibrium, and bi-linear constraint of momentum balance. In the present work, we developed a finite element solver for the latter, inspired by our revisiting of the former in [31]. The nominal solver, as its Lagrangian counterpart, relies on the alternated minimization introduced in the original DDCM [9]. The optimization of the material states is conducted with an efficient tree-based nearest neighbor search algorithm, while the constrained minimization of the mechanical states is achieved from an augmented Lagrangian approach. We successfully assessed our solver with the aid of three sanity check

examples, in two dimensions. In addition, we demonstrated that the initialization of the iterative process is better set to the zero deformation-stress state, as it is equivalent to computing a first linear elastic estimate of stiffness C of the solution to the BVP.

Along with the complete nominal formulation, we investigated the possibility to relax the enforcement of conservation of angular momentum, as it constitutes the only nonlinear constraint to the minimization problem. The solver is then simplified into a so-called “linear” solver, which provides a substantial gain in computational cost. Notwithstanding, its systematic use is subject to the upcoming proof that the consequent error in moment balance is bounded. For example, it should be compared with the error made when the data set is noisy. In this work, we explored a workaround which consists in encoding material-frame indifference in the material data set instead: we enriched material data points with their orbits in $SO(2)$, in an off-line stage. This method did not lower the error in conservation of angular momentum as expected. Another possibility could be to minimize the orbit of the material data points in-line, in the search for the optimal material states, in like manner as the approach presented for geometrically linear elasticity in [9]. Until future investigation completely elucidates this question, we recommend enforcing conservation of angular momentum as a constraint, to avoid misleading results.

The nominal formulation (using $((\mathbf{F}, \mathbf{P}))$) allows for mathematical proofs of convergence, as opposed to the Lagrangian formulation (using (\mathbf{E}, \mathbf{S})). However, numerical restrictions employed here ($p = q = 2$) prevented us from evaluating the constitutive models suggested in [30]. Future work should then focus on assessing the solver with general values of p and q .

Future work could also help improving computational performance, as the current nominal solver is more costly than the Lagrangian solver presented in [31]. **We verified this claim by conducting equivalent Lagrangian simulations to the examples given in Section 3. The computational time is then much lower (approximately 13 times lower, data not shown).** Both approaches require a nonlinear system of algebraic equations to be solved. Yet the size of the nominal system is larger than that of the Lagrangian formulation, as the stress unknowns are more numerous and lie at integration points. **In addition, the Dirichlet boundary conditions could be enforced by direct substitution instead of the Lagrange multiplier and penalty term used herein.**

To conclude, the present work was merely a proof of concept: we provided a valid FE solver for data-driven computational mechanics in finite strain. Future work is focusing on evaluating the solver with more complex material databases, as in [31], and in three-dimensional problems. Indeed, we believe that this formulation is worth exploring further as it paves the way for anelastic large strain formulations, which rely on the multiplicative split of the deformation gradient tensor.

Appendix A. Expressions of the gradient and Hessian matrix of the Augmented Lagrangian function

Appendix A.1. Gradient

The gradient of the augmented Lagrangian function writes

$$\nabla L = \begin{pmatrix} \nabla_u L \\ \nabla_P L \end{pmatrix} \quad (\text{A.1})$$

with

$$\nabla_u L = \nabla_u f + c^D (\nabla_u h^D)^T h^D + c^{\text{req}} (\nabla_u h^{\text{req}})^T h^{\text{req}} - (\nabla_u h^D)^T \lambda - (\nabla_u h^{\text{req}})^T \mu \quad (\text{A.2a})$$

$$\nabla_P L = \nabla_P f + c^{\text{teq}} (\nabla_P h^{\text{teq}})^T h^{\text{teq}} + c^{\text{req}} (\nabla_P h^{\text{req}})^T h^{\text{req}} - (\nabla_P h^{\text{teq}})^T \eta - (\nabla_P h^{\text{req}})^T \mu \quad (\text{A.2b})$$

where the different terms are defined in what follows.

- The gradients $\nabla_u f$ and $\nabla_P f$ of the objective function respectively write

$$[\nabla_u f]_\alpha = \frac{\partial f}{\partial u_{a,i}} = \sum_{e=1}^M w_e C \Delta F_{iJ}^e B_J^{ea}, \quad \forall \alpha(a, i) \quad (\text{A.3a})$$

$$[\nabla_P f]_\iota = \frac{\partial f}{\partial P_{iJ}^e} = w_e C^{-1} \Delta P_{iJ}^e, \quad \forall \iota(e, i, J) \quad (\text{A.3b})$$

with $\Delta \mathbf{F}_e = \mathbf{F}_e - \mathbf{F}_{ie}^*$ and $\Delta \mathbf{P}_e = \mathbf{P}_e - \mathbf{P}_{ie}^*$, and $B_J^{ea} = \frac{\partial \mathcal{N}_a}{\partial X_J}(\mathbf{X}_e)$. The integer α is the global numeration of the displacement array u and ι is the global numeration of the stress array P , related to the local numerations as

$$\alpha(a, i) = (a - 1)n + i \quad (\text{A.4a})$$

$$\iota(e, i, J) = (e - 1)n^2 + (i - 1)n + J \quad (\text{A.4b})$$

with a the node, i, J the directions, e the integration point and n the dimension of the problem.

- The gradient $\nabla_u h^D$ of the Dirichlet boundary conditions write

$$\nabla_u h^D = S \quad (\text{A.5})$$

- The gradient $\nabla_P h^{\text{teq}}$ of the translational equilibrium constraint is defined as

$$\nabla_P h^{\text{teq}} = \bar{B}^T W \quad (\text{A.6})$$

- To express the gradients of the rotational equilibrium constraint, it easier to resort to indicial notation. The gradients $\nabla_u h^{\text{req}}$ and $\nabla_P h^{\text{req}}$ then write

$$[\nabla_u h^{\text{req}}]_{c,\alpha} = \frac{\partial r_{e,i}}{\partial u_{a,j}} = [\epsilon]_{ijk} P_{kL}^e B_L^{ea} = [\epsilon \cdot (\mathbf{P}_e \mathbf{B}_{ea})]_{ij}, \quad \forall c(e, i), \alpha(a, j) \quad (\text{A.7a})$$

$$[\nabla_P h^{\text{req}}]_{c,\iota} = \frac{\partial r_{e,i}}{\partial P_{kL}^e} = [\epsilon]_{ijk} F_{jL}^e = [-\epsilon \cdot \mathbf{F}_e]_{ikL}, \quad \forall c(e, i), \iota(e, k, L) \quad (\text{A.7b})$$

where c is the global numeration of the conservation of angular momentum constraint h^{req} , related to the integration point e and the direction $i \in [1 \dots n_r]$ as

$$c(e, i) = (e - 1)n^2 + i. \quad (\text{A.8})$$

Appendix A.2. Hessian matrix

We proceed to give the expressions of each term in the Hessian matrix

$$\nabla^2 L = \begin{pmatrix} \nabla_u^2 L & \nabla_P \nabla_u L \\ \nabla_u \nabla_P L & \nabla_P^2 L \end{pmatrix} \quad (\text{A.9})$$

of the NR iteration (45).

- The diagonal blocks respectively write

$$\nabla_u^2 L = \nabla_u^2 f + c^D (\nabla_u h^D)^T \nabla_u h^D + c^{\text{req}} (\nabla_u h^{\text{req}})^T \nabla_u h^{\text{req}} \quad (\text{A.10a})$$

$$\nabla_P^2 L = \nabla_P^2 f + c^{\text{req}} (\nabla_P h^{\text{req}})^T \nabla_P h^{\text{req}} + c^{\text{req}} (\nabla_P h^{\text{req}})^T \nabla_P h^{\text{req}} \quad (\text{A.10b})$$

where the hessian blocks of the objective function are defined as

$$[\nabla_u^2 f]_{\alpha_1, \alpha_2} = \frac{\partial^2 f}{\partial u_{b,j} \partial u_{a,i}} = \sum_{e=1}^M w_e C \left(\sum_{K=1}^n B_K^{ea} B_K^{eb} \right) \delta_{ij}, \quad \forall \alpha_1 = \alpha(a, i), \alpha_2 = \alpha(b, j) \quad (\text{A.11a})$$

$$[\nabla_P^2 f]_{\iota_1, \iota_2} = \frac{\partial^2 f}{\partial P_{iJ}^e \partial P_{kL}^e} = w_e C^{-1} \delta_{ik} \delta_{JL}, \quad \forall \iota_1 = \iota(e, i, J), \iota_2 = \iota(e, k, L) \quad (\text{A.11b})$$

where M_{ab} is the number of integration points $e \ni a, b$ connected to both nodes a and b .

- The anti-diagonal blocks of the tangent matrix only involve the gradient and Hessian of the bi-linear constraint h^{req} , as it is the only one coupling u and P . The two blocks are symmetrical to one another such that

$$\nabla_P \nabla_u L = c^{\text{req}} \left((\nabla_u h^{\text{req}})^T \nabla_P h^{\text{req}} + \nabla_P \nabla_u (h^{\text{req}} \cdot h^{\text{req}}) \right) - \nabla_P \nabla_u (\mu \cdot h^{\text{req}}) \quad (\text{A.12a})$$

$$\nabla_u \nabla_P L = (\nabla_P \nabla_u L)^T \quad (\text{A.12b})$$

where the Hessian term of the constraint h^{req} is defined, for any array y of the same size, as

$$[\nabla_P \nabla_u (y \cdot h^{\text{req}})]_{\alpha, \iota} = \sum_{i=1}^{n_r} y_i \epsilon_{ijk} B_L^{ea}, \quad \forall \alpha(a, j), \iota(e, k, L) \quad (\text{A.13})$$

Appendix B. Euclidean mapping

The tree-based search algorithm interface [34] requires the data set and the query points to sit in a standard Euclidean space. The local states $z_e \in \mathcal{Z}_{\text{loc}}^e$, which consist of pairs of second order tensors, must then be recast into a single vector, such that the DD distance between two points in local phase space is equivalent to the Euclidean distance between their respective transformed vectors. In the following, we give the corresponding transformation for the nominal formulation, which relies on the Mandel notation presented in [32]. To reduce the amount of notation the subscript e is omitted in this section.

Appendix B.1. Mandel notation

Let consider a second order tensor $\mathbf{A} \in \mathbb{R}^{n \times n}$, in dimension n . We denote $\bar{\mathbf{A}} \in \mathbb{R}^{n^2}$ the corresponding vector in Mandel notation. For $n = 3$, the components of $\bar{\mathbf{A}}$ are related to \mathbf{A} as follows [32]

$$\bar{\mathbf{A}} = [A_{11} \quad A_{22} \quad A_{33} \quad A_{\underline{23}} \quad A_{\underline{31}} \quad A_{\underline{12}} \quad A_{\underline{32}} \quad A_{\underline{13}} \quad A_{\underline{21}}]^T, \quad (\text{B.1a})$$

with

$$A_{\underline{ij}} = \frac{\sqrt{2}}{2} (A_{ij} + A_{ji}), \quad (\text{B.1b})$$

$$A_{\bar{ij}} = \frac{\sqrt{2}}{2} (A_{ij} - A_{ji}). \quad (\text{B.1c})$$

Note that when \mathbf{A} is symmetrical, then $A_{\bar{ij}} = 0, \forall(i, j)$ and the Mandel vector $\bar{\mathbf{A}}$ can be reduced to its upper part in $\mathbb{R}^{n(n-1)}$. In particular, when $n = 2$, the Mandel notation of $\mathbf{A}_{(2)} \in \mathbb{R}^{2 \times 2}$ writes

$$\bar{\mathbf{A}}_{(2)} = \begin{cases} [A_{11} \quad A_{22} \quad A_{\underline{12}}]^T & \text{if } \mathbf{A}^T = \mathbf{A}, \\ [A_{11} \quad A_{22} \quad A_{\underline{12}} \quad A_{\underline{21}}]^T & \text{otherwise.} \end{cases} \quad (\text{B.2})$$

Appendix B.2. Euclidean mapping

Provided that the C -parameter in the nominal distance is a scalar, the *nominal Euclidean mapping* \mathcal{M} is easily derived. The deformation gradient and first Piola-Kirchhoff stress tensors are simply recast into their factorized Mandel form by

$$\mathcal{M} : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{2n^2} \quad (\text{B.3})$$

$$z = (\mathbf{F}, \mathbf{P}) \mapsto \bar{\mathbf{z}} = \begin{pmatrix} \frac{1}{\sqrt{2}} C^{1/2} \bar{\mathbf{F}} \\ \frac{1}{\sqrt{2}} C^{-1/2} \bar{\mathbf{P}} \end{pmatrix}. \quad (\text{B.4})$$

Thus,

$$V(\mathbf{F}) + V^*(\mathbf{P}) = \bar{\mathbf{z}} \cdot \bar{\mathbf{z}}, \quad (\text{B.5})$$

with V and V^* the quadratic functions defined in (16).

Appendix C. neo-Hookean model for incompressible plane stress problems

The standard incompressible neo-Hookean hyperelastic potential $\Psi(\mathbf{C})$ is given as

$$\Psi(\mathbf{C}) = \frac{1}{2}\mu(\text{tr } \mathbf{C} - 3), \quad (\text{C.1})$$

where $\mathbf{C} = \mathbf{F}^T \mathbf{F} \in \mathbb{R}_{\text{sym}}^{3 \times 3}$ is the right Cauchy-Green stretch tensor and μ is the shear modulus. The second Piola-Kirchhoff stress tensor $\mathbf{S} \in \mathbb{R}_{\text{sym}}^{3 \times 3}$ is expressed as a function of the unknown hydrostatic pressure p [36]:

$$\mathbf{S} = \mu III_C^{-1/3} \left(\mathbf{I} - \frac{1}{3} I_C \mathbf{C}^{-1} \right) + p J \mathbf{C}^{-1}, \quad (\text{C.2})$$

where $I_C = \text{tr } \mathbf{C}$ and $III_C = \det \mathbf{C}$ are the first and third invariants of \mathbf{C} respectively, and $J = \det \mathbf{F}$. When the material is incompressible, $J = 1$, and plane stress conditions are assumed, the hydrostatic pressure can be determined and (C.2) is reduced to a two-dimensional constitutive model.

Specifically, if \mathbf{e}_3 is the plane stress direction, $\mathbf{S} \mathbf{e}_3 = \mathbf{0}$ and the right Cauchy-Green tensor \mathbf{C} writes

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{pmatrix}, \quad (\text{C.3})$$

where, from the incompressibility constraint $J = \sqrt{III_C} = 1$,

$$C_{33} = (\det \mathbf{C}_{(2)})^{-1} \quad (\text{C.4})$$

with $\mathbf{C}_{(2)} \in \mathbb{R}_{\text{sym}}^{2 \times 2}$ the two-dimensional left upper part of \mathbf{C} in (C.3). Together with the plane stress condition $S_{33} = 0$, (C.4) enables the pressure p (C.2) to be explicitly evaluated as

$$p = \frac{1}{3}\mu \left(\text{tr } \mathbf{C}_{(2)} - 2 (\det \mathbf{C}_{(2)})^{-1} \right). \quad (\text{C.5})$$

The in-plane components of the second Piola-Kirchhoff stress tensor are then determined directly by the two-dimensional expression of the neo-Hookean model in plane stress and incompressible conditions as

$$\mathbf{S}_{(2)} = \mu \left(\mathbf{I}_{(2)} - (\det \mathbf{C}_{(2)})^{-1} \mathbf{C}_{(2)}^{-1} \right), \quad (\text{C.6})$$

where the subscript (2) indicates the 2×2 components of a tensor.

References

- [1] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, F. Pierron, Overview of identification methods of mechanical parameters based on full-field measurements, *Experimental Mechanics* 48 (4) (2008) 381–402. doi:10.1007/s11340-008-9148-y.

- [2] F. Pierron, M. Grédiac, Towards Material Testing 2.0. A review of test design for identification of constitutive parameters from full-field measurements, *Strain* 57 (1) (2021) e12370. doi:10.1111/str.12370.
- [3] Ghaboussi J., Garrett J. H., Wu X., Knowledge-based modeling of material behavior with neural networks, *Journal of Engineering Mechanics* 117 (1) (1991) 132–153. doi:10.1061/(ASCE)0733-9399(1991)117:1(132).
- [4] Y. M. A. Hashash, S. Jung, J. Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, *International Journal for Numerical Methods in Engineering* 59 (7) (2004) 989–1005. doi:10.1002/nme.905.
- [5] J. Ling, R. Jones, J. Templeton, Machine learning strategies for systems with invariance properties, *Journal of Computational Physics* 318 (2016) 22–35. doi:10.1016/j.jcp.2016.05.003.
- [6] B. A. Le, J. Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks, *Int. J. Numer. Meth. Engng* 104 (12) (2015) 1061–1084. doi:10.1002/nme.4953.
- [7] V. M. Nguyen-Thanh, L. T. K. Nguyen, T. Rabczuk, X. Zhuang, A surrogate model for computational homogenization of elastostatics at finite strain using high-dimensional model representation-based neural network, *International Journal for Numerical Methods in Engineering* 121 (21) (2020) 4811–4842. doi:10.1002/nme.6493.
- [8] R. Ibañez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta, A manifold learning approach to data-driven computational elasticity and inelasticity, *Archives of Computational Methods in Engineering* 25 (1) (2018) 47–57. doi:10.1007/s11831-016-9197-9.
- [9] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 304 (2016) 81–101. doi:10.1016/j.cma.2016.02.001.
- [10] S. Conti, S. Müller, M. Ortiz, Data-driven problems in elasticity, *Archive for Rational Mechanics and Analysis* 229 (1) (2018) 79–123. doi:10.1007/s00205-017-1214-0.
- [11] A. Leygue, M. Coret, J. Réthoré, L. Stainier, E. Verron, Data-based derivation of material response, *Computer Methods in Applied Mechanics and Engineering* 331 (2018) 184–196. doi:10.1016/j.cma.2017.11.013.
- [12] L. Stainier, A. Leygue, M. Ortiz, Model-free data-driven methods in mechanics: Material data identification and solvers, *Computational Mechanics* 64 (2) (2019) 381–393. doi:10.1007/s00466-019-01731-1.
- [13] J. Réthoré, A. Leygue, M. Coret, L. Stainier, E. Verron, Computational measurements of stress fields from digital images, *International Journal for Numerical Methods in Engineering* 113 (12) (2018) 1810–1826. doi:10.1002/nme.5721.
- [14] A. Leygue, R. Seghir, J. Réthoré, M. Coret, E. Verron, L. Stainier, Non-parametric material state field extraction from full field measurements, *Comput Mech* 64 (2) (2019) 501–509. doi:10.1007/s00466-019-01725-z.
- [15] M. Dalémat, M. Coret, A. Leygue, E. Verron, Measuring stress field without constitutive equation, *Mechanics of Materials* 136 (2019) 103087. doi:10.1016/j.mechmat.2019.103087.
- [16] R. Xu, J. Yang, W. Yan, Q. Huang, G. Giunta, S. Belouettar, H. Zahrouni, T. B. Zineb, H. Hu, Data-driven multiscale finite element method: From concurrence to separation, *Computer Methods in Applied Mechanics and Engineering* 363 (2020) 112893. doi:10.1016/j.cma.2020.112893.
- [17] K. Karapiperis, L. Stainier, M. Ortiz, J. E. Andrade, Data-driven multiscale modeling in mechanics, *Journal of the Mechanics and Physics of Solids* (2020) 104239doi:10.1016/j.jmps.2020.104239.
- [18] T. Kirchdoerfer, M. Ortiz, Data-driven computing with noisy material data sets, *Computer Methods in Applied Mechanics and Engineering* 326 (2017) 622–641. doi:10.1016/j.cma.2017.07.039.
- [19] R. Eggersmann, L. Stainier, M. Ortiz, S. Reese, Model-free data-driven computational mechanics enhanced by tensor voting, *Computer Methods in Applied Mechanics and Engineering* 373 (2021) 113499. arXiv:2004.02503, doi:10.1016/j.cma.2020.113499.
- [20] Y. Kanno, Mixed-integer programming formulation of a data-driven solver in computational elasticity, *Optimization Letters* (Feb. 2019). doi:10.1007/s11590-019-01409-w.
- [21] Y. Kanno, A kernel method for learning constitutive relation in data-driven computational elasticity,

- Japan J. Indust. Appl. Math. (May 2020). doi:10.1007/s13160-020-00423-1.
- [22] Q. He, J.-S. Chen, A physics-constrained data-driven approach based on locally convex reconstruction for noisy database, *Computer Methods in Applied Mechanics and Engineering* 363 (2020) 112791. doi:10.1016/j.cma.2019.112791.
- [23] C. G. Gebhardt, D. Schillinger, M. C. Steinbach, R. Rolfes, A framework for data-driven structural analysis in general elasticity based on nonlinear optimization: The static case, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 112993. doi:10.1016/j.cma.2020.112993.
- [24] C. G. Gebhardt, M. C. Steinbach, D. Schillinger, R. Rolfes, A framework for data-driven structural analysis in general elasticity based on nonlinear optimization: The dynamic case, *International Journal for Numerical Methods in Engineering* 121 (24) (2020) 5447–5468. doi:10.1002/nme.6389.
- [25] T. Kirchdoerfer, M. Ortiz, Data-driven computing in dynamics, *International Journal for Numerical Methods in Engineering* 113 (11) (2018) 1697–1710. doi:10.1002/nme.5716.
- [26] L. T. K. Nguyen, M. Rambauser, M.-A. Keip, Variational framework for distance-minimizing method in data-driven computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 112898. doi:10.1016/j.cma.2020.112898.
- [27] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, M. Ortiz, Model-Free Data-Driven inelasticity, *Computer Methods in Applied Mechanics and Engineering* 350 (2019) 81–99. doi:10.1016/j.cma.2019.02.016.
- [28] P. Carrara, L. De Lorenzis, L. Stainier, M. Ortiz, Data-driven fracture mechanics, *Computer Methods in Applied Mechanics and Engineering* 372 (2020) 113390. doi:10.1016/j.cma.2020.113390.
- [29] L. T. K. Nguyen, M.-A. Keip, A data-driven approach to nonlinear elasticity, *Computers & Structures* 194 (2018) 97–115. doi:10.1016/j.compstruc.2017.07.031.
- [30] S. Conti, S. Müller, M. Ortiz, Data-driven finite elasticity, *Archive for Rational Mechanics and Analysis* (2020). doi:10.1007/s00205-020-01490-x.
- [31] A. Platzer, A. Leygue, L. Stainier, Assessment of data-driven computational mechanics in finite strain elasticity, in: *Constitutive Models for Rubber XI: Proceedings of the 11th European Conference on Constitutive Models for Rubber (ECCMR 2019)*, June 25-27, 2019, Nantes, France, CRC Press, 2019, p. 230.
- [32] R. M. Brannon, *Rotation, Reflection, and Frame Changes, Orthogonal Tensors in Computational Engineering Mechanics*, IOP Publishing, 2018. doi:10.1088/978-0-7503-1454-1.
- [33] J. L. Bentley, Multidimensional binary search trees used for associative searching, *Communications of the ACM* 18 (9) (1975) 509–517. doi:10.1145/361002.361007.
- [34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, *Scikit-learn: Machine learning in python*, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [35] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, Massachusetts, 1996.
- [36] J. Bonet, A. J. Gil, R. D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*, Cambridge University Press, Cambridge, 2016. doi:10.1017/CB09781316336144.