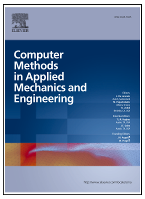
Computer Methods in Applied Mechanics and Engineering 453 (2026) 118803

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma



# A sparse basis for equilibrium stress fields with application for direct data-driven mechanics

Erik Prume[a,*], Chenyi Ji[a,b], Stefanie Reese[a,c], Michael Ortiz[d,e]

[a] Institute of Applied Mechanics, RWTH Aachen University, Mies-van-der-Rohe-Str. 1, D-52074, Aachen, Germany
[b] Institutes of Applied Medical Engineering, RWTH Aachen University, Paulwelsstr. 20, D-52074, Aachen, Germany
[c] University of Siegen, Adolf-Reichwein-Str. 2a, D-57076, Siegen, Germany
[d] Division of Engineering and Applied Science, California Institute of Technology, 1200 E. California Blvd., 91125, Pasadena, CA, USA
[e] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Universitat Politècnica de Catalunya, Jordi Girona 1, 08034, Barcelona, Spain

ARTICLE INFO

*Keywords:*
Data-driven mechanics
Optimization methods
Statistical mechanics
Uncertainty quantification

ABSTRACT

We present a new class of solvers for direct data-driven mechanical problems based on a sparse basis representation of equilibrium stress fields. Our first contribution is an efficient algorithm for computing the required sparse null-space basis on tetrahedral meshes.

Only a single QR decomposition is needed to compute a small remaining set of dense basis vectors associated with boundary conditions and topological holes which can be handled efficiently via a partitioned Cholesky factorization. Building on this, we demonstrate how standard iterative solvers-such as the Newton-Raphson method-can be applied to direct data-driven formulations.

The proposed approach is particularly valuable for challenging problems with complex data distributions requiring systematic exploration of the space of equilibrium stress fields. To this end, we introduce an algorithm that constructs a hierarchical solution set through an eigenvalue decomposition in the joint space of equilibrium stress and compatible strain fields. We demonstrate the proposed methodology with a numerical example involving brittle fracture with probabilistic tensile strength. The resulting family of failure patterns offers valuable insights for uncertainty quantification and design decision-making.

## 1. Introduction

In this paper, we develop a new class of solvers for direct data-driven problems [1,2] by explicitly constructing a basis for equilibrium stress fields on tetrahedral meshes. The basic principle of the data-driven method is to satisfy the equations of kinematic compatibility and equilibrium exactly while minimizing the modeling uncertainties associated with a phenomenological material model. This is achieved by incorporating the data set of stress-strain measurements directly into the formulation of the boundary value problem. Suitable solutions are then defined as the compatible strain fields and equilibrium stress fields that minimize a distance to the data set.

Compatibility can be explicitly enforced by parameterizing the strain fields $\boldsymbol{\varepsilon}$ as

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0 \tag{1}$$

where $\boldsymbol{B}$ is the discrete strain operator, $\boldsymbol{u}$ is the displacement field, and $\boldsymbol{\varepsilon}_0$ is an offset accounting for Dirichlet boundary conditions.

[*] Corresponding author.
*E-mail address:* erik.prume@ifam.rwth-aachen.de (E. Prume).

https://doi.org/10.1016/j.cma.2026.118803
Received 8 December 2025; Received in revised form 29 January 2026; Accepted 3 February 2026
Available online 9 February 2026

In contrast, the equilibrium condition for a stress field $\boldsymbol{\sigma}$ is given by the linear constraint

$$(\boldsymbol{B}^T \boldsymbol{V})\boldsymbol{\sigma} = \boldsymbol{E}\boldsymbol{\sigma} = \boldsymbol{p}. \tag{2}$$

Here, $\boldsymbol{E}$ is the discrete divergence operator, $\boldsymbol{V}$ is a diagonal matrix of material point volumes, and $\boldsymbol{p}$ is the external load vector.

Thus, to derive a basis $\boldsymbol{Z}$ for all equilibrium stress fields we need to compute a basis for the null space, or kernel, of $\boldsymbol{E}$ satisfying

$$\boldsymbol{E}\boldsymbol{Z} = \boldsymbol{0}. \tag{3}$$

Every column of the null basis $\boldsymbol{Z}$ is called a null vector and can be interpreted as a linear independent stress field which is in equilibrium without external forces; it is therefore also called a self-stress field [3]. By taking linear combinations of the null vectors, all possible equilibrium stress fields can be parameterized as

$$\boldsymbol{\sigma} = \boldsymbol{Z}\boldsymbol{v} + \boldsymbol{\sigma}_0 \tag{4}$$

where $\boldsymbol{v}$ represents the degrees of freedom (sometimes referred as redundant forces) and $\boldsymbol{\sigma}_0$ is a particular solution satisfying

$$\boldsymbol{E}\boldsymbol{\sigma}_0 = \boldsymbol{p}. \tag{5}$$

In principle, computing the null basis $\boldsymbol{Z}$ can be simply done by a QR decomposition or singular value decomposition of $\boldsymbol{E}$. However, despite the usual sparsity of $\boldsymbol{E}$, this approach leads to a dense null basis which quickly becomes computationally inefficient for large 3D problems. This difficulty motivated the development of greedy algorithms in the 1980s [4–7] to compute sparse null bases as approximations for the computationally NP-hard problem of finding the sparsest possible null basis [8]. In [9], the relationship to the numerically related problem of matrix sparsification was established.

More recently, [10] has proposed an graph-based algorithm for tetrahedral meshes using numerical factorizations on substructures of the mesh to achieve sparsity of the null basis.

In the first part of this paper we develop an algorithm that efficiently computes a sparse null basis for tetrahedral elements with a very small fraction of dense columns. In particular, we provide an exact count for different types of self-stress fields for simply connected domains and show how they are related in order to avoid linear dependencies. Then, based on the seminal work of [11] for triangular meshes, we derive the null vectors from self-equilibrium force fields defined on the set of edges. Finally, we require only a single QR decomposition in order to compute the remaining, typically very small fraction of null vectors corresponding to indeterminate supports and topological holes. As these null vectors cannot be expected to be very sparse in general, we argue that incorporating a small fraction of dense columns is the most practical approach and show how a partitioned Cholesky factorization can be used for efficient application.

In the second part of the paper, we show how the direct data-driven method can be reformulated as an unconstrained minimization problem using the null basis. In particular, we develop algorithms for two kind of procedures: energy minimization (or exploitation) and exploration of energy landscapes. For energy minimization, we numerically demonstrate the identical convergence behavior of the original fixed-point solver and a modified Newton-Raphson solver for the proposed, unconstrained problem formulation. However, computational gains can only be expected for systems with a very low degree of static indeterminacy. The parameterization of equilibrium stress fields becomes particularly useful for the exploration of multi-modal energy landscapes resulting from non-trivial, noisy data sets where no unique stress-strain relationship can be established, such as in the case of brittle damage [12]. To tackle this challenge, we finally develop a new algorithm based on the Deterministic Annealing framework [13,14] which computes a hierarchical solution set that can be useful for uncertainty quantification purposes.

Finally, we conclude the paper in Section 4 and give an outlook for further applications and developments.

## 2. Part I: An explicit and sparse basis for equilibrium stress fields

### 2.1. Notation

Part I involves the topology and geometry of tetrahedral meshes. We use the following notations. Let $N, E, F, T$ denote the sets of nodes, edges, facets and tetrahedrons, respectively. Let $|\bullet|$ denote the size of a set, for instance $|N|$ denotes the number of vertices. The elements of the sets are labeled by the lower case letters $n \in N, e \in E, f \in F$ and $t \in T$. Furthermore, a geometrical quantity is a set of lower order quantities, for instance the nodes of a tetrahedron $t \in T$ are denoted by $n \in t$. Using the set difference notation, we can identify a facet of $t$ opposite to node $n$ by $(t \setminus n)$. We further use the shorthand notation for conditional sums like $\sum_{n:n\in t} := \sum_{n\in N:n\in t}$ to indicate the sum over all nodes of tetrahedron $t$ and likewise $\sum_{t:n\in t}$ for the sum over all tetrahedrons which share a node $n$. The volumes of various elements are denoted by edge length $l_e$, facet area $A_f$ and tetrahedral volume $V_t$.

### 2.2. Discrete strain and divergence operators.

We consider discrete mechanical problems with linearized kinematics defined on tetrahedral meshes for the three-dimensional case. The basic field quantities involved are the infinitesimal strain field

$$\boldsymbol{\varepsilon} : T \to \mathrm{sym}(\mathbb{R}^3 \times \mathbb{R}^3) \tag{6}$$

and the stress field

$$\boldsymbol{\sigma} : T \to \mathrm{sym}(\mathbb{R}^3 \times \mathbb{R}^3) \tag{7}$$

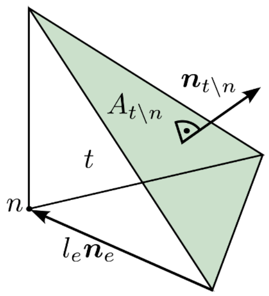


**Fig. 1.** Tetrahedron $t$. the set notation $(t \setminus n)$ denotes the facet of $t$ opposite to node $n$ with area size $A_{t\setminus n}$ and outward unit normal $\boldsymbol{n}_{t\setminus n}$.

which are symmetric tensor fields that are constant over each tetrahedron. We denote the strain and stress in a given tetrahedron $t \in T$ by $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\sigma}_t$.

Both fields are subject to linear constraints: kinematical compatibility and force equilibrium. We enforce compatibility by deriving the strain field as the symmetric gradient of a displacement field

$$\boldsymbol{u} : N \to \mathbb{R}^3 \tag{8}$$

as

$$\boldsymbol{\varepsilon}_t := \frac{1}{2}\left(\mathrm{Grad}_t(\boldsymbol{u}) + \mathrm{Grad}_t^T(\boldsymbol{u})\right). \tag{9}$$

Using a standard barycentric interpolation [15], the displacement gradient reads

$$\mathrm{Grad}_t(\boldsymbol{u}) := -\sum_{n:n\in t} \frac{A_{t\setminus n}}{3V_t} \boldsymbol{u}_n \otimes \boldsymbol{n}_{t\setminus n} \tag{10}$$

where $A_{t\setminus n}$ is the area of the facet of tetrahedron $t$ opposite to node $n$ and $\boldsymbol{n}_{t\setminus n}$ denotes the unit normal of that facet pointing *outwards* of $t$, see Fig. 1 for a visualization. We note that the barycentric interpolation yields the same form as if derived from an isoparametric concept with linear shape functions, but this form proves useful to derive and show further geometric relationships in the following.

Using a suitable Voigt notation

$$\boldsymbol{\varepsilon}_t = (\varepsilon_{t,11}, \varepsilon_{t,22}, \varepsilon_{t,33}, 2\varepsilon_{t,12}, 2\varepsilon_{t,13}, 2\varepsilon_{t,23}) \tag{11}$$

$$\boldsymbol{\sigma}_t = (\sigma_{t,11}, \sigma_{t,22}, \sigma_{t,33}, \sigma_{t,12}, \sigma_{t,13}, \sigma_{t,23}), \tag{12}$$

the strain field can be expressed by the matrix vector product

$$\boldsymbol{\varepsilon} = \boldsymbol{B}^{\mathrm{full}}\boldsymbol{u}^{\mathrm{full}} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0 \tag{13}$$

where the superscript 'full' indicates that the constraint degrees of freedom are included. For a cleaner notation, we do not use a different notation for Voigt and tensor forms but give remarks if required. Rewriting Eq. (13) in Voigt notation, the discrete strain operator consists of the block matrices

$$\boldsymbol{B}^{\mathrm{full}}_{t,n} = -\sum_{n:n\in t} \frac{A_{t\setminus n}}{3V_t} \begin{bmatrix} n_{t\setminus n,1} & 0 & 0 & n_{t\setminus n,2} & n_{t\setminus n,3} & 0 \\ 0 & n_{t\setminus n,2} & 0 & n_{t\setminus n,1} & 0 & n_{t\setminus n,3} \\ 0 & 0 & n_{t\setminus n,3} & 0 & n_{t\setminus n,1} & n_{t\setminus n,2} \end{bmatrix}^T. \tag{14}$$

We say a stress field $\boldsymbol{\sigma}$ is in equilibrium if its nodal divergence balances the external forces $\boldsymbol{p} : N \to \mathbb{R}^3$ as

$$\mathrm{Div}_n(\boldsymbol{\sigma}) + \boldsymbol{p}_n = \boldsymbol{0} \quad \forall n \in N. \tag{15}$$

By Cauchy's stress theorem, we define the divergence

$$\mathrm{Div}_n(\boldsymbol{\sigma}) := \sum_{t:n\in t} \frac{A_{t\setminus n}}{3} \boldsymbol{\sigma}_t \cdot \boldsymbol{n}_{t\setminus n} \tag{16}$$

as the net outward force flux of node $n$.

We call the matrix form of the (negative) divergence operator in Voigt notation the equilibrium matrix $\boldsymbol{E}^{\mathrm{full}}$ consisting of block matrices

$$\boldsymbol{E}^{\mathrm{full}}_{n,t} = V_t(\boldsymbol{B}^{\mathrm{full}}_{t,n})^T \tag{17}$$

noting the adjoint relationship between the strain and divergence operator. Thus, the assembled equilibrium matrix after removing the constraint degrees of freedom can be written as

$$\boldsymbol{E} = \boldsymbol{B}^T\boldsymbol{V} \tag{18}$$

where $\boldsymbol{V}$ is the diagonal matrix of tetrahedral volumes.

### 2.3. Dimensionality and components of the null basis

We start by identifying the number of linear independent null vectors for simply connected domains, i.e. domains without holes. By the rank-nullity theorem, we have

$$\dim(\mathrm{Im}(\boldsymbol{E})) + \dim(\mathrm{Ker}(\boldsymbol{E})) = 6|T| \tag{19}$$

where 'dim' denotes the dimension, 'Im' the image space and 'Ker' the kernel or null space.

Assuming that the supports does not allow any rigid body movements, the dimension of the image space of the reduced equilibrium matrix is

$$\dim(\mathrm{Im}(\boldsymbol{E})) = 3|N| - |D| \tag{20}$$

with $|D|$ being the number of degrees of freedom constraint by Dirichlet boundary conditions. Thus, the dimension of the null space $\boldsymbol{Z}$ of $\boldsymbol{E}$ is

$$\dim(\boldsymbol{Z}) = \dim(\mathrm{Ker}(\boldsymbol{E})) = 6|T| - (3|N| - |D|). \tag{21}$$

We can get further insights by differentiating between internal quantities within the domain and external quantities on the boundary of the domain, indicated by subscripts 'int' and 'ext', respectively. From the counting rules

$$|N| = |N_{\mathrm{int}}| + |N_{\mathrm{ext}}| \tag{22}$$

$$4|T| = 2|F_{\mathrm{int}}| + |F_{\mathrm{ext}}| \tag{23}$$

$$2|E_{\mathrm{ext}}| = 3|F_{\mathrm{ext}}| \tag{24}$$

and Euler's characteristic for closed surfaces

$$|N_{\mathrm{ext}}| - |E_{\mathrm{ext}}| + |F_{\mathrm{ext}}| = 2 \tag{25}$$

we obtain

$$\dim(\boldsymbol{Z}) = 3|F_{\mathrm{int}}| - 3|N_{\mathrm{int}}| + |D| - 6. \tag{26}$$

To derive independent mechanisms, we propose the split

$$\dim(\boldsymbol{Z}) = (3|F_{\mathrm{int}}| - |E_{\mathrm{int}}|) + (|E_{\mathrm{int}}| - 3|N_{\mathrm{int}}|) + (|D| - 6). \tag{27}$$

The motivation for the split is that a group of internal facets is linked by one internal edge, creating a dependency which needs to be resolved, see Fig. 2 a) for a visualization. The consequence is that the following mechanisms are required to produce linear independent null vectors:

- Facet-wise mechanism: $(3|F_{\mathrm{int}}| - |E_{\mathrm{int}}|)$ null vectors. Each of the 3 edges of every internal facet creates one null vector which is supported on the pair of tetrahedrons adjacent to the internal facet. Every internal edge has to be left out *once*.
- Edge-wise mechanism: $(|E_{\mathrm{int}}| - 3|N_{\mathrm{int}}|)$ null vectors. An internal edge left out from the facet-wise mechanism can create one, more complex, null vector supported by the group of tetrahedrons adjacent to the internal edge. For every internal node, three internal edge-wise mechanism has to be left out to ensure linear independent null vectors.
- Dirichlet mechanism: $(|D| - 6)$ null vectors. Every Dirichlet boundary condition creates one null vectors except for 6 supports preventing rigid body motions.
- Holes in the domain induce further null vectors. A general treatment of homologies in the sense of algebraic topology [16] is, however, outside the scope of this paper.

Before we derive the algorithms for each mechanism we establish a useful geometric relation between force fields supported on edges and stress fields as shown next.

### 2.4. Stress fields induced by edge force fields

The preceding analysis suggests that the degrees of freedom are scalars supported on edges. We follow an intuitive picture and think of force fields supported on the edges of each tetrahedron from which the stresses can be computed consistently as shown in this subsection. This approach is a generalization to the three dimensional case of the concepts introduced in [11].

We consider scalar fields

$$P : E \times T \to \mathbb{R} \tag{28}$$

subject to the constraint

$$\sum_{e:n\in e} \left( \sum_{t:e\in t} P_{e,t} \right) (\boldsymbol{x}_{e\setminus n} - \boldsymbol{x}_n)/l_e = \boldsymbol{0} \quad \forall n \in N \tag{29}$$

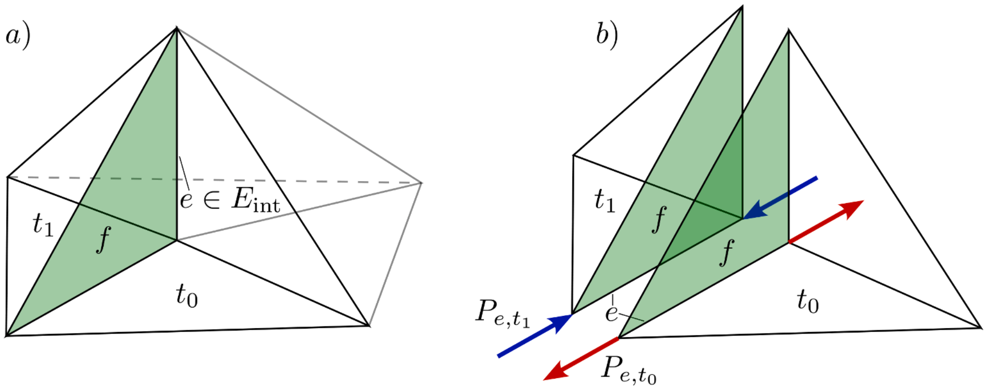


**Fig. 2.** a) Facets adjacent to an internal edge. b) Pair of equal but opposite directing forces along a shared edge of two adjacent tetrahedrons.

where $\boldsymbol{x}_n \in \mathbb{R}^3$ and $\boldsymbol{x}_{e\setminus n} \in \mathbb{R}^3$ are the nodal coordinates of an edge $e$ adjacent to node $n$.

Interpreting $P_e = \sum_{t:e\in t} P_{e,t}$ as the resultant force magnitude of edge $e$, Eq. (29) resembles the classical equilibrium of nodal forces. Then, $P_{e,t}$ can be also seen as the part of $P_e$ distributed to tetrahedron $t$.

We define the associated stress of a tetrahedron $t \in T$ as

$$\boldsymbol{\sigma}_t = \sum_{e:e\in t} \frac{l_e}{V_t} P_{e,t}(\boldsymbol{n}_e \otimes \boldsymbol{n}_e) \tag{30}$$

where $\boldsymbol{n}_e$ is a unit vector along the edge $e$ with an arbitrary but fixed orientation.

By inserting Eq. (30) into the divergence Eq. (16)

$$\mathrm{Div}_n(\boldsymbol{\sigma}) = \sum_{t:n\in t} \frac{A_{t\setminus n}}{3}\left(\sum_{e:e\in t}\frac{l_e}{V_t}P_{e,t}\boldsymbol{n}_e \otimes \boldsymbol{n}_e\right)\cdot \boldsymbol{n}_{t\setminus n} \tag{31}$$

$$= \sum_{t:n\in t}\sum_{e:e\in t} P_{e,t}\frac{(l_e\boldsymbol{n}_e)\cdot(A_{t\setminus n}\boldsymbol{n}_{t\setminus n})}{3V_t}\boldsymbol{n}_e \tag{32}$$

$$= \sum_{e:n\in e}\left(\sum_{t:e\in t} P_{e,t}\right)\mathrm{sign}(\boldsymbol{n}_e\cdot\boldsymbol{n}_{t\setminus n})\boldsymbol{n}_e = \boldsymbol{0} \tag{33}$$

we can show that if a force field $P$ satisfies constraint Eq. (29), then the associated stress field is divergence free. Note in particular that

$$\frac{(l_e\boldsymbol{n}_e)\cdot(A_{t\setminus n}\boldsymbol{n}_{t\setminus n})}{3V_t} = \begin{cases} +1, & \text{if } \boldsymbol{n}_e \text{ points away from } n \\ -1, & \text{if } \boldsymbol{n}_e \text{ points towards } n \\ 0, & \text{if } e \in (t\setminus n) \end{cases} \tag{34}$$

holds (see again Fig. 1).

### 2.5. Facet-wise mechanism

In this section, we give the algorithm how to compute one null vector $\boldsymbol{z}$ based on the facet-wise mechanism. Each null vector corresponds to one internal facet $f \in F_{\text{int}}$ and one of its edges $e \in f$. Along the given edge $e$, a pair of opposite unit forces is applied on the two adjacent tetrahedrons $t_0$ and $t_1$ as

$$P_{e,t_0} = +1 \tag{35}$$

$$P_{e,t_1} = -1 \tag{36}$$

noting that the required constraint (29) is fulfilled trivially since $P_e = P_{e,t_0} + P_{e,t_1} = 0$ and $P_{e,t} = 0$ everywhere else, see Fig. 2 b) for a visualization.

The resulting null vector $\boldsymbol{z}$ is thus supported on $t_0$ and $t_1$ is given by evaluating the stress function (30) to

$$\boldsymbol{z}_{t_0} = +(l_e/V_{t_0})(\boldsymbol{n}_e \otimes \boldsymbol{n}_e) \tag{37}$$

$$\boldsymbol{z}_{t_1} = -(l_e/V_{t_1})(\boldsymbol{n}_e \otimes \boldsymbol{n}_e). \tag{38}$$

We remark that this approach is a direct generalization of the two-dimensional case proposed in [11]. The procedure is summarized in Algorithm 1.

**Algorithm 1** Facet-wise mechanism.

```
Require: Internal facet $f \in F_{int}$
Require: Edge $e \in f$
  $\{t_0, t_1\} \leftarrow \{t \in T : f \in t\}$
  $\boldsymbol{z} \leftarrow \boldsymbol{0}$
  $\boldsymbol{z}_{t_0} \leftarrow +(l_e/V_{t_0})(\boldsymbol{n}_e \otimes \boldsymbol{n}_e)$
  $\boldsymbol{z}_{t_1} \leftarrow -(l_e/V_{t_1})(\boldsymbol{n}_e \otimes \boldsymbol{n}_e)$
  return TensorToVoigt($\boldsymbol{z}$)
```

### 2.6. Edge-wise mechanism

In this section, we derive an algorithm to compute one null vector $\boldsymbol{z}$ based on the edge-wise mechanism.

Every null vector corresponds to one internal edge $e$. Let $T_e = \{t \in T : e \in t\}$ be set set of tetrahedrons adjacent to $e$, which we call a basic module.

Further, let $N_e^{\text{cycle}} = \{n \in N : n \in T_e, n \notin e\}$ be the set of nodes forming a cycle around $e$. We note that in a typical mesh, $N_e^{\text{cycle}}$ contains at least 3 nodes (which is a very well studied case in classical literature [17]) but usually $|N_e^{\text{cycle}}|$ is larger than 3. However, independent of the size of the basic module, the number of edges of a basic module is $3|N_e^{\text{cycle}}| + 1$ which implies that there is exactly one self-stress field for each basic module. In order to derive an edge-tetrahedron force field $P_{e,t}$ satisfying Eq. (29), We implement the following procedure:

1. Compute an equilibrium force field $P_e$ which is only non-zero on the edges of the basic module.
2. Compute $P_{e,t}$ by distributing the edge forces $P_e$ to the tetrahedrons of the basic module.
3. Compute $\boldsymbol{z}$ from $P_{e,t}$ using the stress formula Eq. (30).

In order to obtain a self-equilibrium edge force field, we generalize an algorithm of [18] for basic modules with $|N_e^{\text{cycle}}| \geq 3$. The algorithm exploits the fact that every node in the cycle is shared by exactly four edges with a non-zero force, which is the minimum number of forces in three dimensions to form equilibrium, see Fig. 3 for a visualization. The edge forces can then be determined by a purely geometrical procedure using a recursive application of signed volume ratios as described in the following.

Every node $n \in N_e^{\text{cycle}}$ is joined by the four edges $\boldsymbol{v}_j := (\boldsymbol{x}_{e_j \setminus n} - \boldsymbol{x}_n),\ j \in (1,2,3,4)$ of the basic module with force magnitudes $P_j$.

To identify the ratio between two of the edge forces, $P_1$ and $P_2$ for example, the equation of force equilibrium

$$P_1 \boldsymbol{v}_1/l_1 + P_2 \boldsymbol{v}_2/l_2 + P_3 \boldsymbol{v}_3/l_3 + P_4 \boldsymbol{v}_4/l_4 = \boldsymbol{0} \tag{39}$$

is multiplied by a cross product and contraction of the other two edges as

$$P_1(\boldsymbol{v}_1 \times \boldsymbol{v}_3) \cdot \boldsymbol{v}_4/l_1 + P_2(\boldsymbol{v}_2 \times \boldsymbol{v}_3) \cdot \boldsymbol{v}_4/l_2 = 0 \tag{40}$$

yielding

$$\frac{P_2}{P_1} = -\frac{l_2}{l_1}\frac{(\boldsymbol{v}_1 \times \boldsymbol{v}_3) \cdot \boldsymbol{v}_4}{(\boldsymbol{v}_2 \times \boldsymbol{v}_3) \cdot \boldsymbol{v}_4}. \tag{41}$$

Choosing $P_1 = 1$ as an initial unit force, all other edge forces can be recursively computed by evaluating ratio $P_j/P_1, j \in (2,3,4)$. It is important that edge 1 is an edge of the cycle in order to ensure non-zero denominators as the denominators are then always signed volumes of single tetrahedrons. After repeating this process for all nodes of the cycle in an sorted manner, the force of the central edge $e$ can be computed simply by completing force equilibrium

$$P_e = -\frac{1}{l_e}(\boldsymbol{x}_{e \setminus n} - \boldsymbol{x}_n) \cdot \sum_{e' : n \in e', e' \neq e} \frac{P_{e'}}{l_{e'}}(\boldsymbol{x}_{e' \setminus n} - \boldsymbol{x}_n). \tag{42}$$

at one of its node $n \in e$. For the second step, the edge forces are equally distributed to the tetrahedrons of the basic module as

$$P_{e,t} = \frac{\mathbb{1}(t \in T_e)}{\sum_{t \in T_e} \mathbb{1}(e \in t)} P_e \tag{43}$$

where $\mathbb{1}(\bullet)$ is the indicator function evaluating to 1 if the argument is true and 0 otherwise. We note that the choice (43) is not unique but ensures condition (29). Having computed the force field $P_{e,t}$, the associated null vector follows again from the stress formula (30).

A basic module is visualized in Fig. 3 and the procedure is summarized in Algorithm 2.

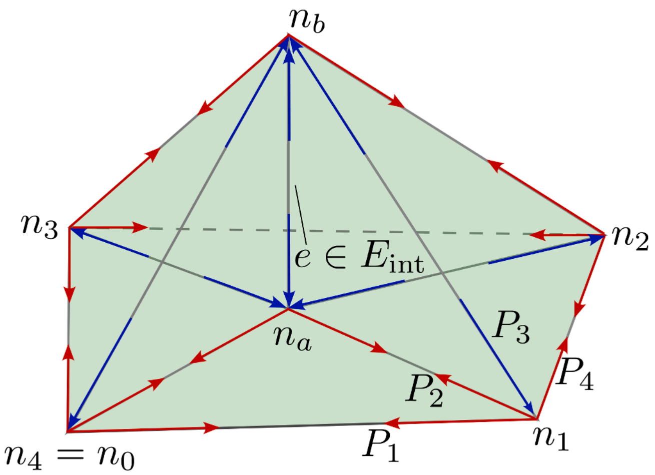


**Fig. 3.** Edge forces of an basic module around an internal edge $e$. The outer nodes form a cycle $(n_1, n_2, n_3, n_4, n_1)$ of length $K = 4$.

**Algorithm 2** Edge-wise mechanism.

```
Require: Internal edge $e \in E_{\text{int}}$
    $z \leftarrow 0$
    $T_e \leftarrow \{t \in T : e \in t\}$
    $\{n_a, n_b\} \leftarrow \{n \in N : n \in e\}$
    $N_e^{\text{cycle}} \leftarrow \{n \in N : n \in T_e\} \setminus \{n_a, n_b\}$
    $(n_1, n_2, \dots, n_K = n_0, n_1) \leftarrow$ Sort($N_e^{\text{cycle}}$) //Sort as an closed cycle using node-edge connectivity
    $P_{(n_0,n_1)} = 1$
    for $k = 1 \dots K$ do
        $P_{(n_k,n_a)} \leftarrow \alpha P_{(n_{k-1},n_k)}$ //$\alpha$ :signed volume ratio using Eq. (41)
        $P_{(n_k,n_b)} \leftarrow \alpha P_{(n_{k-1},n_k)}$
        $P_{(n_k,n_{k+1})} \leftarrow \alpha P_{(n_{k-1},n_k)}$
    end for
    $P_{(n_a,n_b)} \leftarrow$ Eq. (42) // force of central edge $e$
    for all $t \in T_e$ do
        for all $e' \in t$ do
            $D = |\{t \in T_e : e' \in t\}| \in \{1, 2, K\}$//Distribution factor of Eq. (43)
            $P_{e',t} \leftarrow P_{e'}/D$
            $z_t \leftarrow z_t + (l_{e'}/V_t)(P_{e',t})(\boldsymbol{n}_{e'} \otimes \boldsymbol{n}_{e'})$
        end for
    end for
    return TensorToVoigt($z$)
```

As an alternative, one can also consider the related lumped stress method [19], where self-equilibrium force field supported on the edges of triangular meshes is derived from jumps in the gradient of an Airy stress function. However, a generalization to the three-dimensional case is not straightforward [20,21], which is why the present approach is preferred.

### 2.6.1. Minimal example

Before continuing, we provide a minimal example featuring the facet- and edge-wise mechanisms as shown in Fig. 4 A). The facet-wise mechanism contributes $3|F_{\text{int}}| - |E_{\text{int}}| = 3 * 9 - 5 = 22$ null vectors, which are shown as the first 22 columns in the sparsity pattern in Fig. 4 B). Each null vector is supported on 2 adjacent tetrahedrons with 6 stress components each. The edge-wise mechanism contributes $|E_{\text{int}}| - 3|N_{\text{int}}| = 5 - 3 = 2$ null vectors, shown as the last 2 columns in Fig. 4 B). The first null vector of this type cycles around edge $(0, 4)$ supported on 3 tetrahedrons, while the second null vector cycles around edge $(0, 3)$ supported on 4 tetrahedrons. The rank of equilibrium matrix is $\text{rank}(\boldsymbol{E}) = 3|N| - 6 = 12$ since the rigid body motions are not prevented by Dirichlet boundary conditions. Thus, we fulfill the rank-nullity condition (19) by $\text{rank}(\boldsymbol{E}) + \text{rank}(\boldsymbol{Z}) = 12 + 24 = 6|T|$.

## 2.7. Remaining null vectors

Every constrained displacement degree of freedom creates one null vector $z$ of which 6 are linear dependent assuming the boundary conditions do not admit any rigid body motions of the whole body. Furthermore, every hole in the domain introduces additional null

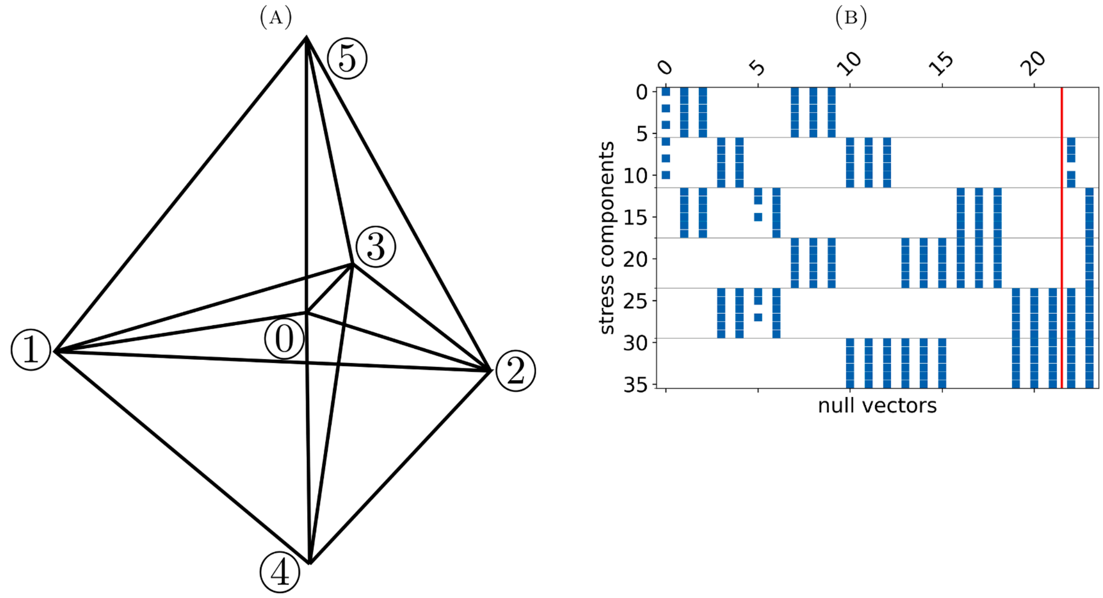


**Fig. 4.** A) Minimal example with 6 tetrahedrons, 1 internal node (labeled 0), 5 internal edges and 9 internal facets. B) Sparsity pattern of the null base $\boldsymbol{Z}$.

vectors. For these remaining null vectors, no closed form solution like the facet-wise or edge-wise mechanism can be generally stated, as they depend on the distribution of the boundary conditions and geometry of holes. In addition, we cannot expect in general highly sparse null vectors as the holes can be quite large or boundary conditions may be sparsely distributed. However, it is important to note that this part of the null space is significantly smaller than the previous parts as the number of boundary conditions scales at most quadratic of the surface and not cubic. Furthermore, the number of holes is not dependent on the discretization and is thus limited. In principle, sparsification strategies can be employed for this part. We have investigated this approach using methods such as least angle regression (LARS) [22] or iteratively reweighted least squares (IRLS) [23]. However, these tests proved uncompetitive due to a significantly larger computational overhead and numerical instabilities such as an ill-conditioned null basis. For our purpose we found that it is sufficient, numerically more robust and more practicable to compute the remaining, dense part of the null base $\boldsymbol{Z}_D$ as the null vectors of equilibrium matrix $\boldsymbol{E}$ which are orthogonal to the existing, sparse null vectors $\boldsymbol{Z}_S$.

In general, an orthogonal (but dense) null basis $\boldsymbol{Z}$ of matrix $\boldsymbol{E}$ can be computed by a sparse QR decomposition [24] of $\boldsymbol{E}^T$ as

$$\boldsymbol{E}^T\boldsymbol{P} = \begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{bmatrix} \tag{44}$$

with column permutation $\boldsymbol{P}$. Then, $\boldsymbol{Q}_2$ is a null basis satisfying $\boldsymbol{E}\boldsymbol{Q}_2 = \boldsymbol{0}$. The reason for this is that any null basis $\boldsymbol{Z}$ also satisfies $\boldsymbol{Z}^T\boldsymbol{E}^T = \boldsymbol{0}$, i.e. the null space of $\boldsymbol{E}$ is orthogonal to the row space of $\boldsymbol{E}$. Then, since $\boldsymbol{Q}_1$ is a basis for the row space of $\boldsymbol{E}$, $\boldsymbol{Q}_2$ spans the null space as it is orthogonal to $\boldsymbol{Q}_1$.

We are now interested in computing only the remaining null vectors of $\boldsymbol{E}$ which are orthogonal to the known part of the null basis $\boldsymbol{Z}_S$. This can be achieved by a QR decomposition of the augmented matrix

$$[\boldsymbol{E}^T, \boldsymbol{Z}_S]\boldsymbol{P} = \begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{bmatrix} \tag{45}$$

from which one obtains $\boldsymbol{Z}_D = \boldsymbol{Q}_2$. The complete null basis is then given by

$$\boldsymbol{Z} = [\boldsymbol{Z}_S, \boldsymbol{Z}_D] \tag{46}$$

where the sparse part $\boldsymbol{Z}_S$ contains the null vectors from the facet-wise and edge-wise mechanisms and $\boldsymbol{Z}_D$ contains all remaining, missing null vectors. It is interesting to note that this mechanism is an instance of the Fault Tolerant Computing [25] theory. If, due to numerical instabilities, $\boldsymbol{Z}_S$ contains some linear dependent null vectors up to a certain precision, the orthogonalization procedure will add all missing null vectors, thus automatically correcting previous errors. In practice, however, we did not encounter this case.

As an alternative, if one is only interested in computing the null vectors corresponding to the Dirichlet boundary conditions, it is numerically more efficient to decompose

$$[\boldsymbol{E}^{\text{full}}\boldsymbol{E}^T]\boldsymbol{P} = \begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{bmatrix} \tag{47}$$

and compute $\boldsymbol{Z}_D = (\boldsymbol{E}^{\text{full}})^T\boldsymbol{Q}_2$. This approach exploits again the orthogonality property of the row space and the null space: since we consider only null vectors for $\boldsymbol{E}$ which are in the row space of $\boldsymbol{E}^{\text{full}}$, we ensure that those are orthogonal to its null basis $\boldsymbol{Z}_S$.

The algorithms are summarized in Algorithm 3 and Algorithm 4. While Algorithm 3 computes *all* remaining null vectors, Algorithm 4 computes only null vectors from Dirichlet boundary conditions in a more efficient way.

**Algorithm 3** Remaining null vectors.

```
Require: Equilibrium matrix $\boldsymbol{E}$
Require: Known part of the null base $\boldsymbol{Z}_S$
  QR decomposition with column ordering: $[\boldsymbol{E}^T, \boldsymbol{Z}_S]\boldsymbol{P} = [\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2]\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{bmatrix}$
  $\boldsymbol{Z}_D \leftarrow \boldsymbol{Q}_2$
  return $\boldsymbol{Z}_D$
```

**Algorithm 4** Dirichlet null vectors (Alternative to Alg. (3)).

```
Require: Full and reduced equilibrium matrices $\boldsymbol{E}^{\text{full}}, \boldsymbol{E}$
  QR decomposition with column ordering: $[\boldsymbol{E}^{\text{full}}\boldsymbol{E}^T]\boldsymbol{P} = [\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2]\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{bmatrix}$
  $\boldsymbol{Z}_D \leftarrow (\boldsymbol{E}^{\text{full}})^T\boldsymbol{Q}_2$
  return $\boldsymbol{Z}_D$
```

### 2.8. Sparse null basis assembly algorithm

In this section, we evaluate the previously developed mechanisms while correctly handling the conditions for linear dependencies as derived in Section 2.3. While it is in principle possible to compute the edge-wise and facet-wise part of the null basis separately, the combined evaluation has the advantage that no column sorting is required as the sparse matrix is automatically also banded. In addition, since all null vectors are linear independent by construction, we do not required an additional factorization in order to remove redundant null vectors. The final algorithm is given in Algorithm 5 .

**Algorithm 5** Assembly algorithm.

```
  $\boldsymbol{Z}_S \leftarrow \boldsymbol{0}$
  $\mathrm{I}_e \leftarrow$ false $\forall e \in E_{\text{int}}$
  $\mathrm{I}_e \leftarrow$ true $\forall e \in E_{\text{ext}}$
  $\mathrm{I}_n \leftarrow 0 \ \forall n \in N_{\text{int}}$
  $\mathrm{I}_n \leftarrow 3 \ \forall n \in N_{\text{ext}}$
  $i \leftarrow 1$
  for all $f \in F_{\text{int}}$ do
      for all $e \in f$ do
          if $\mathrm{I}_{\mathrm{e}}$ = false then
              $\mathrm{I}_e \leftarrow$ true   //skip every internal edge once
              $\{n_a, n_b\} \leftarrow \{n \in N : n \in e\}$
              if $\mathrm{I}_{\mathrm{n_a}} < 3$ then
                  $\mathrm{I}_{n_a} \leftarrow \mathrm{I}_{n_a} + 1$   //skip every internal node three times
              else if $\mathrm{I}_{\mathrm{n_b}} < 3$ then
                  $\mathrm{I}_{n_b} \leftarrow \mathrm{I}_{n_b} + 1$   //skip every internal node three times
              else
                  $\boldsymbol{Z}_{S,i} \leftarrow$ Edge-wise-mechanism($e$) (Alg. 2)
                  $i \leftarrow i + 1$
              end if
          else
              Facet-wise mechanism
              $\boldsymbol{Z}_{S,i} \leftarrow$ Facet-wise-mechanism($f, e$) (Alg. 1)
              $i \leftarrow i + 1$
          end if
      end for
  end for
  $\boldsymbol{Z}_D \leftarrow$ RemainingColumns($\boldsymbol{E}^{\text{full}}, \boldsymbol{E}, \boldsymbol{Z}_S$) (Alg. 3 or Alg. 4)
  return $\boldsymbol{Z}_S, \boldsymbol{Z}_D$
```

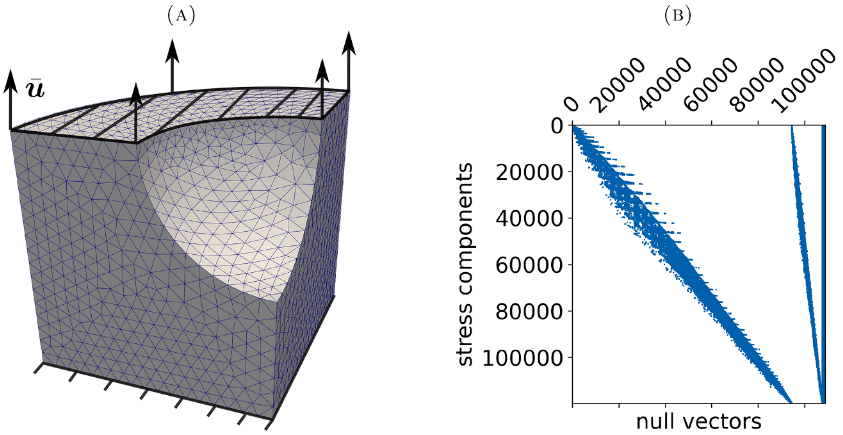


**Fig. 5.** A) Benchmark example boundary value problem. B) Sparsity pattern of the null base. For visualization, the edge-wise and facet-wise parts are plotted separately.

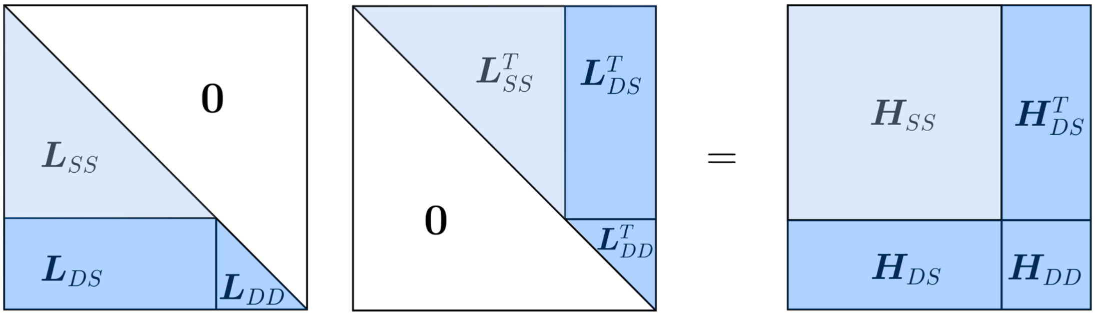


**Fig. 6.** Partitioned Cholesky factorization $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{H}$. Light blue and dark blue blocks are sparse and dense matrices, respectively.

### 2.8.1. *Benchmark example*

In order to demonstrate how the proposed algorithm performs with a larger problem, we consider the mesh shown in Fig. 5 A). The mesh consists of 19,954 elements and 4030 nodes. The displacements of the nodes at the top and bottom sides are prescribed in all directions, removing in total 1299 displacement degrees of freedoms. The required topological connectivities are obtained from the open-source finite element library dolfinx [26].

The sparsity pattern of the null base shown in Fig. 5 B) visualizes that the facet-wise mechanism (first block of columns) is by far the largest contribution with 94,373 columns, followed by the edge-wise mechanism (middle block) with 13,267 columns. Lastly, the Dirichlet boundary conditions contribute 1299 dense columns (last block) which are approximately 1.2% of all columns. The sparsity, as measured by the number of nonzero elements divided by the matrix dimensions, of the sparse part (facet-wise and edge-wise mechanism) is with less than 0.012% very low. The computation times are promising: a Python implementation running on an AMD Ryzen 9 7950X 16-Core Processor CPU requires  4 seconds for the sparse part and  3 seconds for the dense part using Alg. (4). The error of the null basis as measured by the norm $\|\boldsymbol{E}\boldsymbol{Z}\|_2 \approx 10^{-12}$ is at machine precision.

From a practical point of view, forming large dense matrices has to be avoided has much as possible. For instance, $\boldsymbol{Q}_2$ should be computed by an application of Householder transformations (as provided by the libraries) on the last 1299 columns of an identity matrix instead of forming the whole matrix $\boldsymbol{Q}$ explicitly.

## 2.9. *Efficient Cholesky factorization of the null basis*

As elaborated in the previous sections, we consider a null basis $\boldsymbol{Z} = [\boldsymbol{Z}_S, \boldsymbol{Z}_D]$ consisting of a sparse part $\boldsymbol{Z}_S$ and a dense part $\boldsymbol{Z}_D$ which has much less columns than the sparse part. For the later application we require factorizations of $\boldsymbol{H} = \boldsymbol{Z}^T\boldsymbol{D}\boldsymbol{Z}$ where $\boldsymbol{D}$ is a symmetric block diagonal matrix. While $\boldsymbol{Z}^T\boldsymbol{Z}$ is indeed a sparse and banded matrix due to the orthogonality of $\boldsymbol{Z}_D$, a matrix $\boldsymbol{D}$ destroys this property in general leading to an unfavorable sparsity pattern of $\boldsymbol{H}$.

In this subsection, we will briefly present a *partitioned* Cholesky factorization which stores $\boldsymbol{Z}_D$ separately as a dense matrix leading to much more efficient factorization and solving operations. As visualized in Fig. 6, matrix $\boldsymbol{H}$ consists of the large, sparse block $\boldsymbol{H}_{SS} = \boldsymbol{Z}_S^T\boldsymbol{D}\boldsymbol{Z}_S$ and the small, dense blocks $\boldsymbol{H}_{DS} = \boldsymbol{Z}_D^T\boldsymbol{D}\boldsymbol{Z}_S$, $\boldsymbol{H}_{DS}^T$ and $\boldsymbol{H}_{DD} = \boldsymbol{Z}_D^T\boldsymbol{D}\boldsymbol{Z}_D$.

The main block $\boldsymbol{L}_{SS}$ of the lower triangular matrix $\boldsymbol{L}$ can then be efficiently computed by a sparse, supernodal Cholesky decomposition [27] as $\boldsymbol{L}_{SS}\boldsymbol{L}_{SS}^T = \boldsymbol{H}_{SS}$. Using this factorization, we can solve $\boldsymbol{L}_{SS}\boldsymbol{L}_{DS}^T = \boldsymbol{H}_{DS}^T$ for $\boldsymbol{L}_{DS}^T$. Finally, $\boldsymbol{L}_{DD}$ is obtained by a dense but small Cholesky factorization $\boldsymbol{L}_{DD}\boldsymbol{L}_{DD}^T = \boldsymbol{H}_{DD} - \boldsymbol{L}_{DS}\boldsymbol{L}_{DS}^T$.

Then, for solving $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{b}$, the forward and backward substitution of $\boldsymbol{L}$ can also be carried out separately for the sparse and dense parts. The necessary steps are summarized in Algorithm 6.

**Algorithm 6** Partitioned Cholesky factorization and solving.

```
Require: Sparse columns $\boldsymbol{Z}_S$, dense columns $\boldsymbol{Z}_D$
Require: Symmetric block diagonal matrix $\boldsymbol{D}$
Require: Right-hand side $\boldsymbol{b}$

   #Factorization
   $\boldsymbol{H}_{SS} \leftarrow \boldsymbol{Z}_S^T\boldsymbol{D}\boldsymbol{Z}_S$
   $\boldsymbol{H}_{DS} \leftarrow \boldsymbol{Z}_D^T\boldsymbol{D}\boldsymbol{Z}_S$
   $\boldsymbol{H}_{DD} \leftarrow \boldsymbol{Z}_D^T\boldsymbol{D}\boldsymbol{Z}_D$
   $\boldsymbol{L}_{SS} \leftarrow \texttt{SparseCholesky}(\boldsymbol{H}_{SS})$
   $\boldsymbol{L}_{DS}^T \leftarrow \boldsymbol{L}_{SS}^{-1}\boldsymbol{H}_{DS}^T$
   $\boldsymbol{L}_{DD} \leftarrow \texttt{DenseCholesky}(\boldsymbol{H}_{DD} - \boldsymbol{L}_{DS}\boldsymbol{L}_{DS}^T)$

   #Solving
   $[\boldsymbol{b}_S, \boldsymbol{b}_D] \leftarrow \boldsymbol{b}$
   $\boldsymbol{y}_S = \boldsymbol{L}_{SS}^{-1}(\boldsymbol{b}_S)$
   $\boldsymbol{y}_D = \boldsymbol{L}_{DD}^{-1}(\boldsymbol{b}_D - \boldsymbol{L}_{DS}\boldsymbol{y}_S)$
   $\boldsymbol{x}_D \leftarrow \boldsymbol{L}_{DD}^{-T}\boldsymbol{y}_D$
   $\boldsymbol{x}_S \leftarrow \boldsymbol{L}_{SS}^{-T}(\boldsymbol{y}_S - \boldsymbol{L}_{DS}^T\boldsymbol{x}_D)$

   return $[\boldsymbol{x}_S, \boldsymbol{x}_D]$
```

## 3. Part II: Application for direct data-driven mechanics

### 3.1. Background: Direct data-driven mechanics

We consider compatible and equilibrium states

$$(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) \in \mathbb{R}^{6|T|} \times \mathbb{R}^{6|T|} \qquad \text{s.t.} \qquad \boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0 \qquad \text{and} \qquad \boldsymbol{B}^T\boldsymbol{V}\boldsymbol{\sigma} = \boldsymbol{p} \tag{48}$$

of mechanical system with $|T|$ material points. Furthermore, we assume to have a data set

$$D = \{(\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i')\}_{i=1}^{n_D} \quad , \quad (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i') \in \mathbb{R}^6 \times \mathbb{R}^6 \tag{49}$$

of $n_D$ stress-strain pair measurements. We assume a homogeneous material behavior, i.e. every material point shares the same material data set.

As proposed by [1], the feasibility of an admissible state can be directly quantified based on the available data by the sum of nearest neighbor distances

$$F(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) = \sum_{t=1}^{|T|} V_t \min_{(\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i') \in D} \text{dist}_t^2((\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t), (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i')) \tag{50}$$

in each material point. The local distances are defined as

$$\text{dist}_t^2((\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t), (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i')) = (\boldsymbol{\varepsilon}_i' - \boldsymbol{\varepsilon}_t) \cdot \mathbb{C}_t(\boldsymbol{\varepsilon}_i' - \boldsymbol{\varepsilon}_t) + (\boldsymbol{\sigma}_i' - \boldsymbol{\sigma}_t) \cdot \mathbb{C}_t^{-1}(\boldsymbol{\sigma}_i' - \boldsymbol{\sigma}_t) \tag{51}$$

with metric coefficients $\mathbb{C}_t$ and material point volumes $V_t$. As proposed in [2,12], this rather strict measure can be relaxed but by the weighted distance to all data points as given by the energy

$$\mathcal{E}_\beta(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) = -\frac{1}{\beta} \sum_{t=1}^{|T|} \log\left( \sum_{i=1}^{n_D} \exp(-\beta V_t \text{dist}_t^2((\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t), (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i'))) \right) \tag{52}$$

with a regularization parameter $\beta$. Notably, the original distance measure (50) can be recovered by the limit

$$\lim_{\beta\to\infty} \mathcal{E}_\beta(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) = F(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) \tag{53}$$

reducing the sum over all data points to only the closest data point, which we will refer in the following as the 'min-dist' limit.

With this at hand, the direct data-driven problem can be written as the constraint double minimization problem

$$\min_{\boldsymbol{u}, \boldsymbol{\sigma}} \mathcal{E}_\beta(\boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0, \boldsymbol{\sigma}) \qquad \text{s.t.} \qquad \boldsymbol{B}^T\boldsymbol{V}\boldsymbol{\sigma} = \boldsymbol{p}. \tag{54}$$

The solution of this problem is thus the compatible strain and equilibrium stress field which minimizes the weighted distance to measured stress-strain pairs in all material points. In [1], a fixed-point iteration was proposed to solve the 'min-dist' limit case which was extended in [2] to solve the more general, 'min-energy' case. Therein, an annealing scheme is used which slowly increases $\beta \to \infty$ in order to converge to the global minima. For each smoothness level $\beta$, a fixed-point iteration is applied.

The fixed point iteration consists of the two operations of data search and projection to admissible states and is repeated until convergence is achieved. The data search is a local operation. Given an admissible, local state $(\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t)$, we define the generalized data search as

$$\mathbb{E}_D[\boldsymbol{\varepsilon}_t'] = \sum_{i=1}^{n_D} p_i(\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t)\boldsymbol{\varepsilon}_i' \tag{55}$$

$$\mathbb{E}_D[\boldsymbol{\sigma}_t'] = \sum_{i=1}^{n_D} p_i(\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t)\boldsymbol{\sigma}_i' \tag{56}$$

with the softmax function

$$p_i(\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t) = \frac{\exp(-\beta V_t \mathrm{dist}_t^2((\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t), (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i')))}{\sum_{i=1}^{n_D} \exp(-\beta V_t \mathrm{dist}_t^2((\boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}_t), (\boldsymbol{\varepsilon}_i', \boldsymbol{\sigma}_i')))}. \tag{57}$$

The name data search comes from the fact that this operation reduces in the 'min-dist' limit to a nearest neighbor search which can be very efficiently performed, see [28] for a comparison in this context.

The projection step computes the closest compatible strain field to $\boldsymbol{\varepsilon}' := \{\mathbb{E}_D[\boldsymbol{\varepsilon}_t']\}_{t=1}^{|T|}$ and the closest equilibrium stress field to $\boldsymbol{\sigma}' := \{\mathbb{E}_D[\boldsymbol{\sigma}_t']\}_{t=1}^{|T|}$. This can be efficiently performed by solving two decoupled system of equations

$$(\boldsymbol{B}^T \boldsymbol{V} \mathbb{C} \boldsymbol{B})\boldsymbol{u} = \boldsymbol{B}^T \boldsymbol{V} \mathbb{C} \boldsymbol{\varepsilon}' \tag{58}$$

$$(\boldsymbol{B}^T \boldsymbol{V} \mathbb{C} \boldsymbol{B})\boldsymbol{\eta} = \boldsymbol{p} - \boldsymbol{B}^T \boldsymbol{V} \boldsymbol{\sigma}' \tag{59}$$

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0 \tag{60}$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' + \mathbb{C}\boldsymbol{B}\boldsymbol{\eta} \tag{61}$$

with $\mathbb{C} = \mathrm{diag}(\mathbb{C}_t)$. Hereby, $\boldsymbol{\eta}$ are Lagrange multipliers enforcing the equilibrium constraint, see [1,2] for further details. Finally, we note that several solver variations has been proposed in the literature. For better accuracy, [29] proposed a randomized variant of the data search step, while [30] considered an adaptive scheme for the metric coefficients. An alternative approach is also the reformulation as a mixed-integer problem [31]. Particularly noteworthy is also the inverse problem formulation [32] which can be used to obtain data sets from digital image correlation experiments. Further strategies for adaptive, goal-oriented data acquisition were presented in [33].

### 3.2. Reformulation using null basis

The availability of a sparse basis for equilibrium stress fields allows us to formulate new solver schemes for the class of direct data-driven problems using the parameterization

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0 \tag{62}$$

$$\boldsymbol{\sigma} = \boldsymbol{Z}\boldsymbol{v} + \boldsymbol{\sigma}_0. \tag{63}$$

With this at hand, we can rewrite problem (54) as an unconstrained minimization problem

$$\min_{\boldsymbol{u},\boldsymbol{v}} \mathcal{E}_\beta(\boldsymbol{B}\boldsymbol{u} + \boldsymbol{\varepsilon}_0, \boldsymbol{Z}\boldsymbol{v} + \boldsymbol{\sigma}_0) \tag{64}$$

which can be minimized by standard algorithms like the Newton-Raphson method. In particular, we can derive explicit expressions for the Jacobian and Hessian of $\mathcal{E}_\beta$ with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$ as shown in the following. The Jacobian reads

$$\boldsymbol{J}_u(\boldsymbol{u}, \boldsymbol{v}) = \frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{u}} = \boldsymbol{B}^T \frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{B}^T \mathrm{diag}\left(\frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{\varepsilon}_t}\right) \tag{65}$$

$$\boldsymbol{J}_v(\boldsymbol{u}, \boldsymbol{v}) = \frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{v}} = \boldsymbol{Z}^T \frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{\sigma}} = \boldsymbol{Z}^T \mathrm{diag}\left(\frac{\partial \mathcal{E}_\beta}{\partial \boldsymbol{\sigma}_t}\right) \tag{66}$$

with

$$\frac{\partial \mathcal{E}_\beta(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})}{\partial \boldsymbol{\varepsilon}_t} = -2V_t \mathbb{C}_t (\mathbb{E}_D[\boldsymbol{\varepsilon}_t'] - \boldsymbol{\varepsilon}_t) \tag{67}$$

$$\frac{\partial \mathcal{E}_\beta(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}_t} = -2V_t \mathbb{C}_t^{-1} (\mathbb{E}_D[\boldsymbol{\sigma}_t'] - \boldsymbol{\sigma}_t). \tag{68}$$

Similarly, the Hessian is given by

$$\boldsymbol{H}_u(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{B}^T \mathrm{diag}\left(\frac{\partial^2 \mathcal{E}_\beta}{\partial \boldsymbol{\varepsilon}_t^2}\right) \boldsymbol{B} \tag{69}$$

$$\boldsymbol{H}_v(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{Z}^T \operatorname{diag}\left(\frac{\partial^2 \mathcal{E}_\beta}{\partial \boldsymbol{\sigma}_t^2}\right)\boldsymbol{Z} \tag{70}$$

with

$$\frac{\partial^2 \mathcal{E}_\beta}{\partial \boldsymbol{\varepsilon}_t^2} = 2V_t\mathbb{C}_t - 4\beta V_t^2 \mathbb{C}_t \left[\sum_{i=1}^{n_D} p_i(\boldsymbol{\varepsilon}_t,\boldsymbol{\sigma}_t)(\boldsymbol{\varepsilon}_i' \otimes \boldsymbol{\varepsilon}_i') - \mathbb{E}_D[\boldsymbol{\varepsilon}_t'] \otimes \mathbb{E}_D[\boldsymbol{\varepsilon}_t']\right]\mathbb{C}_t \tag{71}$$

$$\frac{\partial^2 \mathcal{E}_\beta}{\partial \boldsymbol{\sigma}_t^2} = 2V_t\mathbb{C}_t^{-1} - 4\beta V_t^2 \mathbb{C}_t^{-1} \left[\sum_{i=1}^{n_D} p_i(\boldsymbol{\varepsilon}_t,\boldsymbol{\sigma}_t)(\boldsymbol{\sigma}_i' \otimes \boldsymbol{\sigma}_i') - \mathbb{E}_D[\boldsymbol{\sigma}_t'] \otimes \mathbb{E}_D[\boldsymbol{\sigma}_t']\right]\mathbb{C}_t^{-1}. \tag{72}$$

Thus, the Hessian matrices consist of the constant terms

$$\bar{\boldsymbol{H}}_u = 2\boldsymbol{B}^T\boldsymbol{V}\mathbb{C}\boldsymbol{B} \tag{73}$$

$$\bar{\boldsymbol{H}}_v = 2\boldsymbol{Z}^T\boldsymbol{V}\mathbb{C}^{-1}\boldsymbol{Z} \tag{74}$$

and a remaining term capturing the covariance around an equilibrium state $(\boldsymbol{u},\boldsymbol{v})$. Thus, we can write

$$\boldsymbol{H}_u(\boldsymbol{u},\boldsymbol{v}) = \bar{\boldsymbol{H}}_u - \beta \operatorname{Cov}_u(\boldsymbol{u},\boldsymbol{v}) \tag{75}$$

$$\boldsymbol{H}_v(\boldsymbol{u},\boldsymbol{v}) = \bar{\boldsymbol{H}}_v - \beta \operatorname{Cov}_v(\boldsymbol{u},\boldsymbol{v}). \tag{76}$$

The availability of the Jacobian and Hessian with respect to the degrees of freedom $\boldsymbol{u}$ and $\boldsymbol{v}$ allows us to use standard solver schemes like the Newton-Raphson scheme to solve Problem (64) as shown in Algorithm 7. However, for sufficiently large $\beta$, the Hessian is not positive semidefinite in general. Thus, the problem is non-convex and the solution of a minimization scheme depends on the initial values. Therefore, one should also use a *modified* Newton-Raphson scheme which involves only the constant parts $\bar{\boldsymbol{H}}_u$ and $\bar{\boldsymbol{H}}_v$ in order to guarantee positive definiteness. Interestingly, using only the constant part of the Hessian, the Newton-Raphson update step $-\bar{\boldsymbol{H}}^{-1}\boldsymbol{J}(\boldsymbol{u},\boldsymbol{v})$ resembles exactly one fixed point iteration of data search and projection as shown numerically in the next example.

**Algorithm 7** (Modified) Newton-Raphson solver.

```
Require: Geometry $\boldsymbol{B}, \boldsymbol{Z}$
Require: Load vectors $\boldsymbol{\varepsilon}_0, \boldsymbol{\sigma}_0$
Require: Regularization parameter $\beta$
Require: Initial solution $(\boldsymbol{u},\boldsymbol{v})$
  Factorize $\bar{\boldsymbol{H}}_u = \boldsymbol{L}_u\boldsymbol{L}_u^T, \bar{\boldsymbol{H}}_v = \boldsymbol{L}_v\boldsymbol{L}_v^T$
  while not converged do
    $\boldsymbol{u} \leftarrow \boldsymbol{u} - \bar{\boldsymbol{H}}_u^{-1}\boldsymbol{J}_u(\boldsymbol{u},\boldsymbol{v})$
    $\boldsymbol{v} \leftarrow \boldsymbol{v} - \bar{\boldsymbol{H}}_v^{-1}\boldsymbol{J}_v(\boldsymbol{u},\boldsymbol{v})$
  end while
  $(\boldsymbol{\varepsilon},\boldsymbol{\sigma}) \leftarrow (\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\varepsilon}_0, \boldsymbol{Z}\boldsymbol{v}+\boldsymbol{\sigma}_0)$
  return $(\boldsymbol{u},\boldsymbol{v}), (\boldsymbol{\varepsilon},\boldsymbol{\sigma})$
```

### 3.2.1. *Numerical example: Newton-Raphson and fixed-point iteration*

We compare the proposed Newton-Raphson solver with the original fixed-point iteration solver numerically with the boundary value defined in section (2.8.1). We consider the 'min-dist' limit with $\beta \to \infty$. In this case, the covariance terms in the Hessian vanishes, thus, the constant approximation of the Hessian becomes exact. At the top nodes, a (dimensionless) displacement of $\bar{\boldsymbol{u}} = (0, 1, 0)$ is prescribed. We assume a linear elastic material behavior with elastic stiffness matrix $\boldsymbol{C}$ and sample 99,770 data points normally distributed around the reference solution. Furthermore, the metric coefficients are chosen to $\mathbb{C}_t = \boldsymbol{K}$.

In Fig. 7 the convergence of the energy and computation times for both solver schemes are reported. Remarkably, both solvers show exactly the same convergence after starting at the same, randomly chosen, equilibrium state. The solving time of approximately 12 seconds is still satisfactory given the size of the boundary value problem and data set.

However, for this example, the solving time and the memory requirements of the Newton-Raphson solver are significantly higher than the original fixed-point solver [1] since a bigger system of equation is solved. Thus, the fixed-point iteration is generally preferable if available. However, the Newton-Raphson solver might be a better choice for systems with a very low degree of static indeterminacy, or in situations when a general purpose, or 'black-box' solver is required for specific implementations.

## 3.3. *Hierarchical solution set: Deterministic annealing*

As previously studied in [2,12], the energy landscape $\mathcal{E}_\beta(\boldsymbol{u},\boldsymbol{v})$ becomes usually very rough for increasing values of $\beta$ , i.e. forming multiple local minima. Moreover, these minima can occur at different scales of $\beta$. At low temperatures (large $\beta$), individual data points start to resolve, causing many local minima scattered around each data point. However, local minima can also occur at higher temperatures due to structural effects and well separated clusters within local data sets. This motivates us to develop an algorithm which can systematically identify all, or at least multiple, local minima at each temperature.

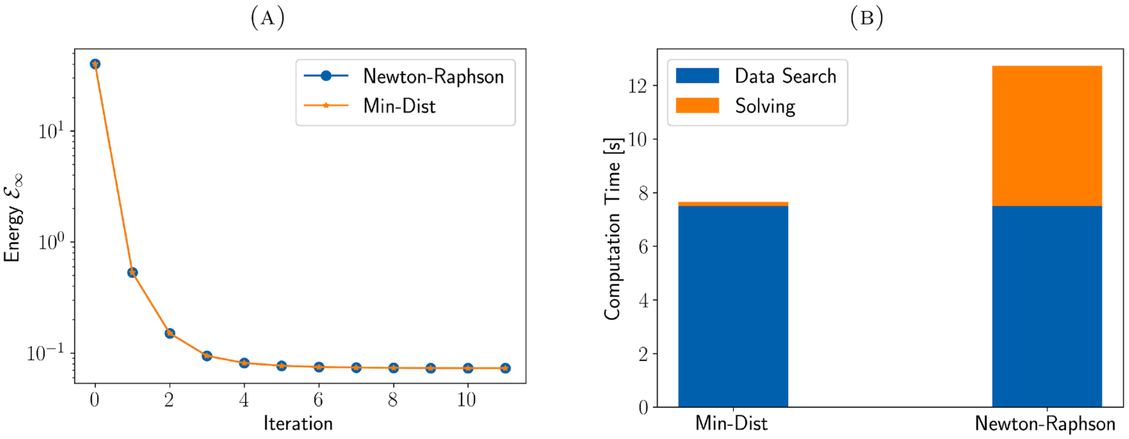


**Fig. 7.** Numerical comparison of the Newton-Raphson algorithm and the original fixed-point iteration 'min-dist' solver [1]. A) Convergence of the energy B) Computational times.

Following the framework of Deterministic Annealing [13,14], the key insight is that local minima occur if an eigenvalue of the Hessian matrix becomes zero. Furthermore, the associated eigenvector carries the information in which direction the local minima are separated. In physical analogy, this process is also called a phase transformation at a critical (inverse) temperature $\beta_{\text{crit}}$.

Since the Hessian matrix of $\mathcal{E}_\beta$ consists of two separated matrices $\boldsymbol{H}_u$ and $\boldsymbol{H}_v$ (Eqs. 75 and 76), we need to solve two generalized eigenvalue problems in the form of

$$\det(\bar{\boldsymbol{H}}_k - \beta \text{Cov}_k) = 0 \tag{77}$$

with $k \in \{u, v\}$. Using the Cholesky factorization of the constant term $\bar{\boldsymbol{H}}_k = \boldsymbol{L}_k \boldsymbol{L}_k^T$, this is equivalent to the standard eigenvalue problem

$$\det(\boldsymbol{I} - \beta \boldsymbol{L}_k^{-1} \text{Cov}_k \boldsymbol{L}_k^{-T}) = 0. \tag{78}$$

This determines the lowest value of $\beta$ at which a phase transformation occurs as

$$\beta_{\text{crit}} = \min\left(\frac{1}{\lambda_{u,\max}}, \frac{1}{\lambda_{v,\max}}\right) \tag{79}$$

where $\lambda_{k,\max}$ is the largest eigenvalue of $\boldsymbol{L}_k^{-1}\text{Cov}_k(\boldsymbol{u}, \boldsymbol{v})\boldsymbol{L}_k^{-T}$.

Furthermore, the direction of the split is given by

$$\boldsymbol{r}_u = \begin{cases} \boldsymbol{0}, & \lambda_{u,\max} < \lambda_{v,\max} \\ \boldsymbol{L}_u^{-T} \boldsymbol{w}_{u,\max}, & else \end{cases} \tag{80}$$

$$\boldsymbol{r}_v = \begin{cases} \boldsymbol{0}, & \lambda_{v,\max} < \lambda_{u,\max} \\ \boldsymbol{L}_v^{-T} \boldsymbol{w}_{v,\max}, & else \end{cases} \tag{81}$$

where $\boldsymbol{w}_{k,\max}$ is the eigenvector associated to the largest eigenvalue $\lambda_{k,\max}$. We remark that the eigenvalue problem is defined on the Hessian with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$, i.e. directly in the space of kinematical compatible strains and equilibrium stress fields. In particular, this step cannot be replaced by a projection type operation, making the availability of the basis $\boldsymbol{Z}$ crucial.

From a practical point of view, one can note that solving for the largest eigenvalue of a symmetric matrix can be efficiently performed without forming the full matrix, for instance by Lanczos algorithm [34]. Nevertheless, the sparsity of $\boldsymbol{B}$ and $\boldsymbol{Z}$ and the partitioned Cholesky factorization (Section 2.9) is important for a good performance.

Since the covariance matrix itself depends on $\beta$, Eq. (79) is an implicit equation. Thus, the critical temperatures cannot be directly computed, but has to be estimated by an annealing schedule which slowly increases $\beta$. In each iteration, or quench, every solution state $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{S}$ of the current solution set $\mathcal{S}$ is updated by a deterministic minimization scheme at the current temperature. For this step, either the fixed-point iteration [2] or the Newton-Raphson algorithm may be used. Then, the maximum eigenvalue problem is solved and checked if the actual value of $\beta$ exceeds $\beta_{\text{crit}}$. If this is the case, the solution is duplicated and perturbed along the principal axis as $(\boldsymbol{u} \pm \gamma_u \boldsymbol{r}_u, \boldsymbol{v} \pm \gamma_v \boldsymbol{r}_v)$ by a certain perturbation amplification $\gamma_u$ and $\gamma_v$. We note that the $\gamma$ parameters should be chosen sufficiently large to ensure that the new solutions do not converge again into the same local minima. The final algorithm is summarized in Algorithm 8.

**Algorithm 8** Deterministic annealing.

```
Require: Geometry $\boldsymbol{B}, \boldsymbol{Z}$
Require: Load vectors $\boldsymbol{\varepsilon}_0, \boldsymbol{\sigma}_0$
Require: Number of quenches $N_Q$, maximum number of solutions $K_{\max}$
Require: Initial $\beta_0$ and final $\beta_{\max}$
Require: Perturbation parameters $\gamma_u, \gamma_v$
  $\mathcal{S} \leftarrow \{(\boldsymbol{u}_0, \boldsymbol{v}_0)\}$//random initialization of solution set
  $\alpha \leftarrow (\beta_{\max}/\beta_0)^{1/N_Q}$
  $\beta \leftarrow \beta_0$
  $K \leftarrow 1$
  for $q = 1 \dots N_Q$ do
    for $i = 1 \dots |\mathcal{S}|$ do
      $\mathcal{S}_i \leftarrow \text{argmin}\ \mathcal{E}_\beta(\mathcal{S}_i)$//Newton-Raphson or fixed-point iteration
      $\mathcal{S}' \leftarrow \{\}$
      if $K < K_{\max}$ then
        $(\boldsymbol{u}, \boldsymbol{v}) \leftarrow \mathcal{S}_i$
        $\beta_{\text{crit}}, \boldsymbol{r}_u, \boldsymbol{r}_v \leftarrow$ MaxEvProblem($\boldsymbol{u}, \boldsymbol{v}, \beta$) // Eqs. (79–81)
        if $\beta \geq \beta_{\text{crit}}$ then
          $\mathcal{S}_i \leftarrow (\boldsymbol{u} - \gamma_u \boldsymbol{r}_u, \boldsymbol{v} - \gamma_v \boldsymbol{r}_v)$
          $\mathcal{S}' \leftarrow \mathcal{S}' \cup (\boldsymbol{u} + \gamma_u \boldsymbol{r}_u, \boldsymbol{v} + \gamma_v \boldsymbol{r}_v)$
          $K \leftarrow K + 1$
        end if
      end if
    end for
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$
    $\beta \leftarrow \alpha\beta$
  end for
  $\beta \leftarrow \beta_{\max}$
  for $i = 1 \dots K$ do
    $\mathcal{S}_i \leftarrow \text{argmin}\ \mathcal{E}_\beta(\mathcal{S}_i)$
  end for
  return $\mathcal{S}$
```

The algorithm is, as the name suggests, purely deterministic, which constitutes a key difference from the algorithm presented in the previous work [12]. In that work, a probabilistic Monte Carlo sampling scheme based on the related concept of simulated annealing was employed. Therein, random moves are proposed, and the inverse temperature $\beta$ controls the probability of accepting moves with increasing energy. Notably, the proposed parametrization of equilibrium stress fields allows for the implementation of more efficient, gradient-based Monte Carlo sampling schemes, such as Hamiltonian Monte Carlo [35]. However, our tests showed that the method of deterministic annealing proved more efficient in finding low-energy states within complex, multi-modal energy landscapes.

### *3.3.1. Numerical example: Probabilistic failure of a plate with holes*

We revisit the simple, 1-dimensional probabilistic failure example studied in [12] in a more complex, 3-dimensional setting. We consider the boundary value problem shown in Fig. 8 A) and assume a linear elastic material behavior with a Weibull-distributed Drucker-Prager stress tensile strength as

$$\sigma_{\text{DP}} = \frac{1}{2}(1 - \sigma_t/\sigma_c)\text{tr}(\boldsymbol{\sigma}) + \frac{1}{2}(1 + \sigma_t/\sigma_c)\sqrt{\frac{1}{2}\boldsymbol{\sigma}_D : \boldsymbol{\sigma}_D} \tag{82}$$

$$P_{\text{fail}}(\sigma_{\text{DP}}) = 1 - e^{-(\sigma_{\text{DP}}/\sigma_t)^k} \tag{83}$$

where $\boldsymbol{\sigma}_D$ is the stress deviator $\boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\boldsymbol{I}$ and 'tr' denotes the trace of a matrix.

Specifically, the artificial data set was created using the following procedure: 1) Compute the linear elastic solution of the boundary value problem, giving $|T|$ data points. 2) Add a normally distributed noise term to each strain state. 3) Compute the stress components by linear elasticity. 4) Draw $|T|$ Weibull-distributed tensile strengths. 5) For each data point, set the stress components to zero if its Drucker-Prager stress exceeds the tensile strength. 4) Optional: repeat the process.

The following parameters were chosen: elastic parameters $E = 100$ and $\nu = 0.3$; tensile and compression strength $\sigma_t = 10$ and $\sigma_c = 100$ (positive per definition); Weibull shape parameter $k = 10$; noise term with standard deviation $s = 0.01$. In total, 3 iterations of data generations were performed, resulting in $3|T| = 10563$ data points. The resulting distribution of Drucker-Prager stresses of the data set and the sampled tensile strengths are visualized in Fig. 8 B).

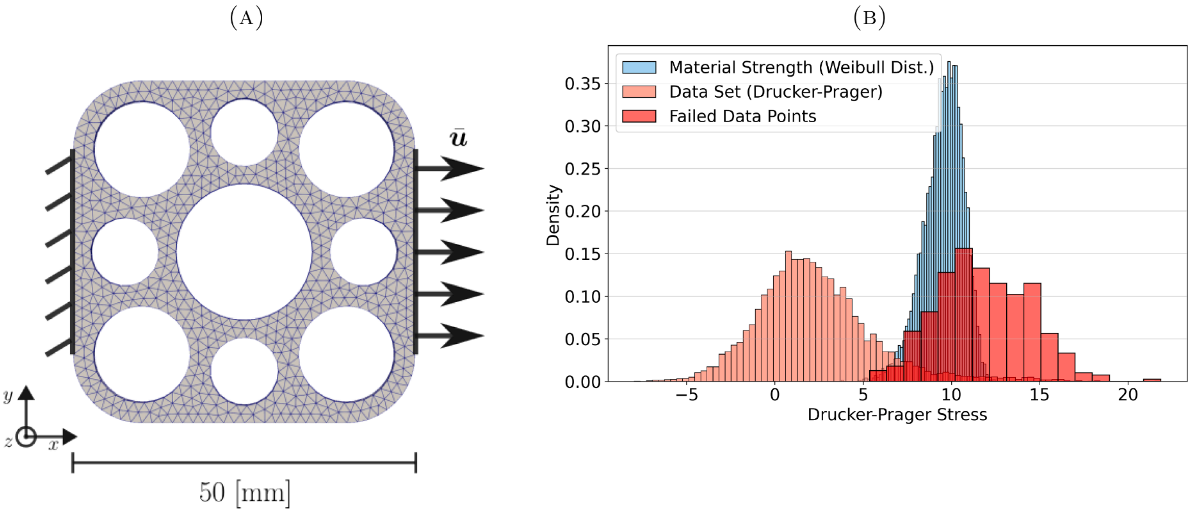


**Fig. 8.** A) Plate with holes boundary value problem. B) Drucker-Prager stress distribution of the local data set. The failure probability is weibull distributed.

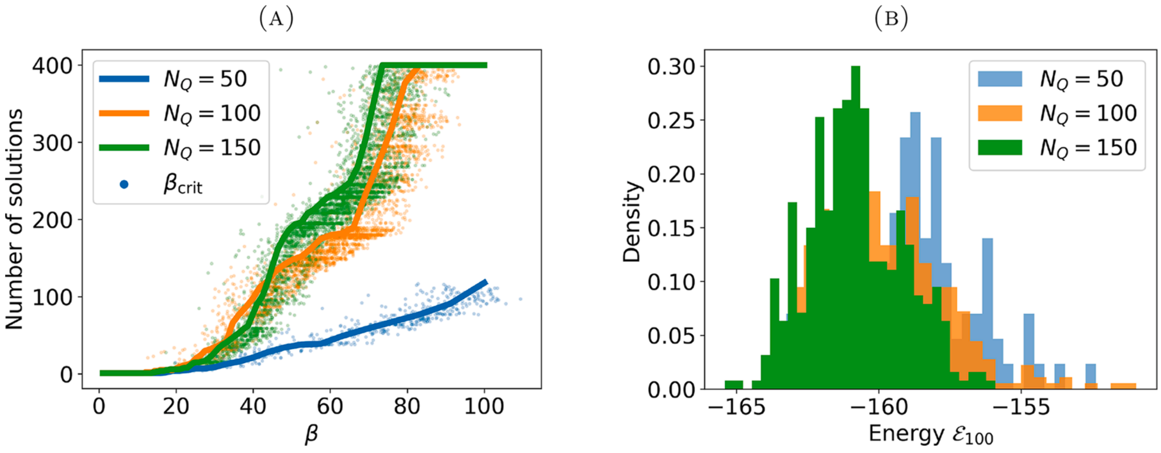


**Fig. 9.** A) Evolution of the number of solutions during the annealing procedure with critical temperatures for different annealing rates. The maximum number of solution is set to 400. B) Energy distribution of the final solution set for different annealing rates.

The sparse basis for equilibrium stress fields has been computed using the algorithms from Part 1 of this paper with the general Algorithm 3 since the domain contains holes. The total dimension of $\boldsymbol{Z}$ is 16896, containing 14,619 facet-wise and 1977 edge-wise null vectors as well as 252 null vectors from Dirichlet boundary conditions and 48 null vectors from the holes.

First, we study the influence of the annealing schedule. Following [12], we estimate the order of magnitude of $\beta_{\max}$ with $\beta_{\max}^{-1}$ proportional to the average nearest neighbor distance of each data point in the data set times the average material point volume size. Based on this, we choose conservatively $\beta_{\max} = 100$. We further choose the perturbation parameters $\gamma_u = 10|\boldsymbol{u}|/|\boldsymbol{r}_u|$ and $\gamma_v = 10|\boldsymbol{v}|/|\boldsymbol{r}_v|$.

In Fig. 9 A), we show the evolution of the number of solutions for different number of quenches $N_Q \in \{50, 100, 150\}$ with a maximum number of allowed solution $K_{\max} = 400$. In addition, we show for each solution its critical inverse temperature $\beta_{\text{crit}}$ in each iteration. Thus, a new solution is generated whenever a point $\beta_{\text{crit}}$ is below the actual $\beta$ of the iteration (that is, on the left side of the line). If the annealing speed is too fast ($N_Q = 50$, blue line), many possible phase transitions are not detected and the solution space is not sufficiently explored. For slower annealing speeds ($N_Q \in \{100, 150\}$), we observe a more exhaustive development of solutions. Furthermore, we can observe two distinct regions of phase transformations: the first one for approximately $\beta \in [20, 60]$ where phase transitions occur at a relatively high temperature and the second one for lower temperatures $\beta > 70$ with an exponential increase in solutions resolving finer details of the data set.

The following computation times for the different number of quenches were recorded in a Python implementation: 32 min ($N_Q = 50$), 103 min ($N_Q = 100$), 134 min ($N_Q = 150$). The critical operations are the energy, gradient and Hessian computations, since the summation is over all data points in all material points. We have used an NVIDIA GeForce RTX 4090 to perform the embarrassingly parallel summation in a brute force manner. For larger data sets, it may be beneficial to use data search algorithms to include only

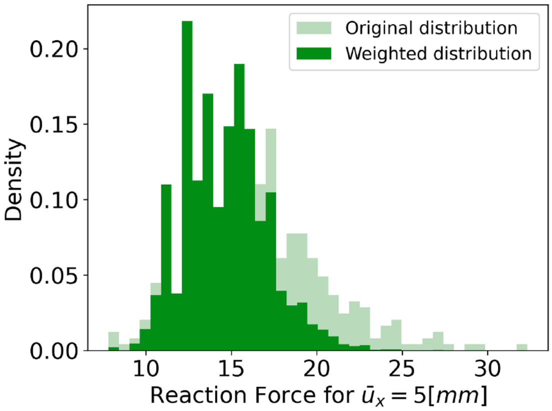


**Fig. 10.** Original and energy-weighted reaction force distribution of 400 solutions at a given displacement loading step after $N_Q = 150$ quenches.

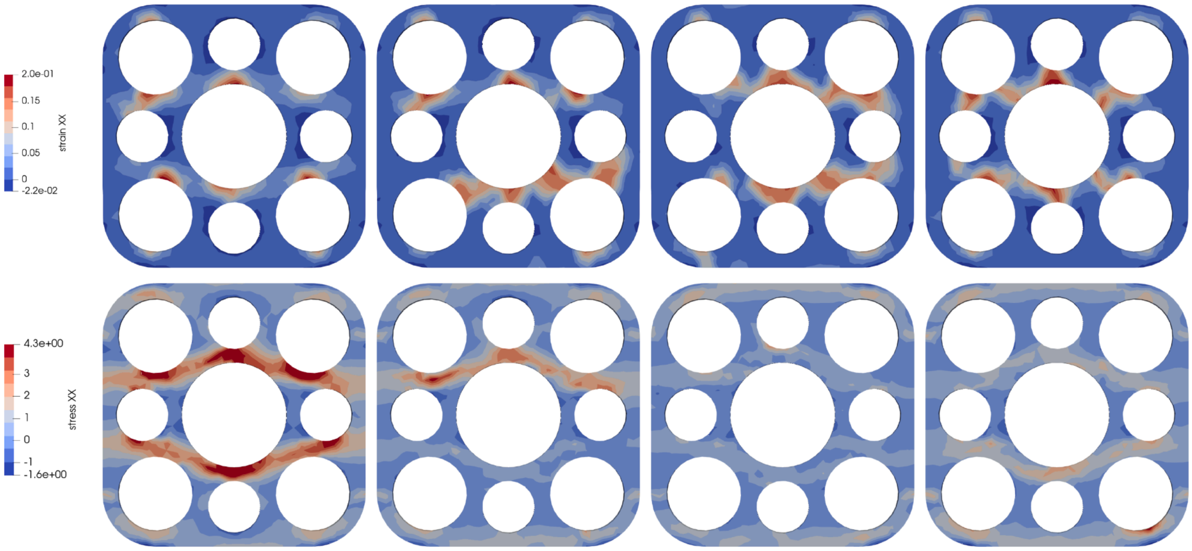


**Fig. 11.** A selection of solutions for the same boundary value problem highlighting the different failure possibilities. Top row: Strain $\varepsilon_{xx}$ fields, bottom row: Stress $\sigma_{xx}$ fields. From left to right: Large reaction force (≈ 25 [N]), moderate reaction force (≈ 12 [N]), low reaction force (≈ 8 [N]), minimum energy solution.

relevant data points in the summation, as previously studied in [12,28]. Furthermore, the algorithm can benefit from a parallel implementation as each solution evolves independently from each other.

In Fig. 9 B), the energy distributions of the solution sets at the end of the annealing schedule for the three different annealing speeds are shown. The important conclusion is that all solutions have an energy in the same order of magnitude, thus there is not a single solution as an unique minimizer. Nevertheless, the energy of every solution can be directly used to quantify how close the equilibrium solution state $(\boldsymbol{u}, \boldsymbol{v})$ is to the given data set $D$. Using the histogram re-weighting method [36], the information about the relative energy differences can also be used to evaluate the results in an energy-weighted manner by computing the weights

$$w_k = e^{-\Delta\beta(\mathcal{E}_{\beta,k} - \mathcal{E}_{\beta,\min})} \tag{84}$$

where $\Delta\beta$ is a factor which can be used to control how strictly energy differences should be handled relative to the solution with minimum energy $\mathcal{E}_{\beta,\min}$. In Fig. 10, the reaction force distribution of the solution set after $N_Q = 150$ quenches is shown unweighted (light green) as well weighted by the energy differences according to Eq. (84) with $\Delta\beta = 1$. The reaction forces are computed by summing $(\boldsymbol{B}^{\text{full}})^T\boldsymbol{\sigma}$ at the degrees of freedom in $x$-direction at the clamped node set, see Fig. 8 A).

Finally, we show in Fig. 11 the strain and stress fields in $x$-direction for a selection of solutions after $N_Q = 150$ iterations. We emphasize that the boundary value problem is for every solution the same, but due to the probabilistic tensile strength and thus noisy data set, no unique stress-strain relationship exists resulting in a large variety of possible outcomes.

The visualization shows that every solution represents one possible crack pattern, giving important information for uncertainty quantification and design decisions. Even though the local data set is relatively simple, the possible outcomes can become, due to the global equilibrium and compatibility constraints, quite complex. We argue that in such cases with non-trivial variability in the stresses, the consistent parametrization of the space of equilibrium stress fields is particularly useful since the eigenvalue analysis, and thus the solution space exploration, can include all information about data, compatibility and equilibrium.

## 4. Conclusions

In this paper, we have developed a novel class of solvers for direct data-driven mechanics using a sparse basis for equilibrium stress fields. First, we developed an algorithm capable of computing a sparse null basis of the equilibrium matrix for large three-dimensional problems within seconds. A detailed analysis of the different components of the null basis enabled us to derive linearly independent generator mechanisms based on self-equilibrium force fields. Furthermore, a partitioned Cholesky factorization was presented to efficiently process the small fraction of dense columns related to indeterminate supports and holes.

Based on this foundation, we have demonstrated how to use standard solvers, such as a Newton-Raphson algorithm, to solve the data-driven problem. However, except for system with a low degree of static indeterminacy, the approach is computationally less efficient than the original fixed point solver. Nevertheless, it might be useful for situations where the implementation requires the use of an unconstrained solver. Another potential application might be situations when no system of equations should be solved, since any gradient descent scheme can be directly used.

We conclude that the true strength of this approach lies in problems with complex and non-trivial data sets requiring an consistent exploration of the space of equilibrium stress fields. The characteristic of such problems are rough, multi-modal energy landscapes which are difficult to explore. As an example, we have presented a brittle fracture problem with probabilistic failure strength. An eigenvalue analysis in the space of equilibrium stress fields and compatible strain fields enabled an effective exploration of the complex solution space. As a result, different possible failure modes were discovered giving important information for uncertainty quantification and design processes.

This paper introduces several research questions and possibilities for future work. First, an interesting question from the field of algebraic topology [16] is how to analyze the topological homologies to precisely determine the number of null vectors associated with holes. Furthermore, as an alternative to the proposed practical approach of using dense columns for the supports and holes, developing techniques to find sparse representations could be considered.

A further important open research question concerns the locking behavior of the proposed method. Standard tetrahedron element formulations that utilize linear displacement interpolation often exhibit artificial stiffness when approaching the incompressible limit. However, as noted by [37], the data-driven formulation bears a close relation to mixed finite element formulations, which are generally known to demonstrate much more favorable locking behaviors. Consequently, further investigations are required to fully characterize this aspect. Additionally, it would be interesting to investigate whether the presented concepts could be adapted to other element types, such as hexahedral elements.

The approach of computing a solution set using deterministic annealing can be applied to different data distributions and boundary value problems. To achieve this, developing robust choices for parameters like annealing speed, maximum solution set size, and perturbation sizes is crucial. Further research is required to develop efficient data sampling strategies that cover all relevant states of deformation. This is particularly important for path-dependent materials, for which the response depends on the loading trajectory [29]. We believe that many applications, particularly safety-critical ones, benefit from the much deeper insights provided by the proposed approach.

### CRediT authorship contribution statement

**Erik Prume:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Chenyi Ji:** Investigation; **Stefanie Reese:** Supervision, Resources, Project administration, Methodology, Funding acquisition; **Michael Ortiz:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

### Data availability

Data will be made available on request.

### Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

EP and SR gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) through the project “Multi-scale modeling based on a novel combination of direct data-driven methods with Fourier transform-based microstructure simulation” (RE 1057/58-1, project number 532163998). MO gratefully acknowledges the financial support of the  Centre Internacional de Mètodes Numèrics a l’Enginyeria (CIMNE) of the Universitat Politecnica de Catalunya (UPC), Spain, through the UNESCO Chair in Numerical Methods in Engineering.

## References

[1] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, Comp. Meth. Appl. Mech. Eng. 304 (2016) 81–101. https://linkinghub.elsevier.com/retrieve/pii/S0045782516300238, https://doi.org/10.1016/j.cma.2016.02.001

[2] T. Kirchdoerfer, M. Ortiz, Data driven computing with noisy material data sets, Comp. Meth. Appl. Mech. Eng. 326 (2017) 622–641. https://linkinghub.elsevier.com/retrieve/pii/S0045782517304012, https://doi.org/10.1016/j.cma.2017.07.039

[3] S. Pellegrino, C.R. Calladine, Matrix analysis of statically and kinematically indeterminate frameworks, Int. J. Solids Struct. 22 (4) (1986) 409–428.

[4] M.T. Heath, R.J. Plemmons, R.C. Ward, et al., Sparse orthogonal schemes for structural optimization using the force method, SIAM J. Sci. Stat. Comp. 5 (3) (1984) 514–532. http://epubs.siam.org/doi/10.1137/0905038, https://doi.org/10.1137/0905038

[5] J.R. Gilbert, M.T. Heath, Computing a sparse basis for the null space, SIAM J. Alg. Disc. Meth. 8 (3) (1987) 446–459.

[6] A. Pothen, Sparse null basis computations in structural optimization, Num. Math. 55 (5) (1989) 501–519.

[7] A. Kaveh, Suboptimal cycle bases of graphs for the flexibility analysis of skeletal structures, Comp. Meth. Appl. Mech. Eng. 71 (3) (1988) 259–271.

[8] T.F. Coleman, A. Pothen, The null space problem i. complexity, SIAM J. Alg. Disc. Meth. 7 (4) (1986) 527–537. https://epubs.siam.org/doi/10.1137/0607059, https://doi.org/10.1137/0607059

[9] L.-A. Gottlieb, T. Neylon, Matrix sparsification and the sparse null space problem, in: International Workshop on Randomization and Approximation Techniques in Computer Science, Springer, 2010, pp. 205–218.

[10] A. Kaveh, M.S. Massoudi, Efficient finite element analysis using graph-theoretical force method; tetrahedron elements, Int. J. Civ. Engin. 12 (2) (2014).

[11] B. Fraeijs de Veubeke, Upper and lower bounds in matrix structural analysis, Rapp. du LTAS (1963).

[12] E. Prume, S. Reese, M. Ortiz, et al., Model-free data-driven inference in computational mechanics, Comp. Meth. Appl. Mech. Eng. (2023) 19.

[13] K. Rose, E. Gurewitz, G.C. Fox, et al., Statistical mechanics and phase transitions in clustering, Phys. Rev. Lett. 65 (8) (1990) 945–948. https://link.aps.org/doi/10.1103/PhysRevLett.65.945, https://doi.org/10.1103/PhysRevLett.65.945

[14] K. Rose, Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, Proc. IEEE 86 (11) (1998) 2210–2239. http://ieeexplore.ieee.org/document/726788/, https://doi.org/10.1109/5.726788

[15] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method for Solid and Structural Mechanics, Elsevier, 2005.

[16] J.R. Munkres, Elements of Algebraic Topology, CRC press, 2018.

[17] J.C. Maxwell, XLV. *On reciprocal figures and diagrams of forces*, London, Edinburgh, Dublin Philos. Mag. J. Sci. 27 (182) (1864) 250–261. https://www.tandfonline.com/doi/full/10.1080/14786446408643663, https://doi.org/10.1080/14786446408643663

[18] O. Aloui, J. Flores, D. Orden, L. Rhode-Barbarigos, et al., Cellular morphogenesis of three-dimensional tensegrity structures, Comp. Meth. Appl. Mech. Eng. 346 (2019) 85–108. https://linkinghub.elsevier.com/retrieve/pii/S0045782518305814, https://doi.org/10.1016/j.cma.2018.10.048

[19] F. Fraternali, M. Angelillo, A. Fortunato, et al., A lumped stress method for plane elastic problems and the discrete-continuum approximation, Int. J. Solids Struct. 39 (25) (2002) 6211–6240. https://linkinghub.elsevier.com/retrieve/pii/S0020768302004729, https://doi.org/10.1016/S0020-7683(02)00472-9

[20] M. Konstantatou, P. D'Acunto, A. McRobie, et al., Polarities in structural analysis and design: n-dimensional graphic statics and structural transformations, Int. J. Solids Struct. 152–153 (2018) 272–293. https://linkinghub.elsevier.com/retrieve/pii/S0020768318302786, https://doi.org/10.1016/j.ijsolstr.2018.07.003

[21] M. Hablicsek, M. Akbarzadeh, Y. Guo, et al., Algebraic 3D graphic statics: reciprocal constructions, Comput.-Aided Des. 108 (2019) 30–41. https://linkinghub.elsevier.com/retrieve/pii/S0010448518303026, https://doi.org/10.1016/j.cad.2018.08.003

[22] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression (2004).

[23] R. Chartrand, Wotao Yin, Iteratively reweighted algorithms for compressive sensing, in: 2008Thinspace IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, Las Vegas, NV, USA, 2008, pp. 3869–3872. ISSN: 1520–6149, http://ieeexplore.ieee.org/document/4518498/, https://doi.org/10.1109/ICASSP.2008.4518498

[24] T.A. Davis, Algorithm 915, suitesparseQR: multifrontal multithreaded rank-revealing sparse QR factorization, ACM Transp. Math. Softw. (TOMS) 38 (1) (2011) 1–22.

[25] A. Arora, M. Gouda, Closure and convergence: a foundation of fault-tolerant computing, IEEE Trans. Softw. Eng. 19 (11) (1993) 1015–1027.

[26] I.A. Baratta, J.P. Dean, J.S. Dokken, M. Habera, J. HALE, C.N. Richardson, M.E. Rognes, M.W. Scroggs, N. Sime, G.N. Wells, DOLFINx: the next generation FEnics problem solving environment (2023).

[27] Y. Chen, T.A. Davis, W.W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, ACM Transp. Math. Softw. (TOMS) 35 (3) (2008) 1–14.

[28] R. Eggersmann, L. Stainier, M. Ortiz, S. Reese, et al., Efficient data structures for model-free data-driven computational mechanics, Comp. Meth. Appl. Mech. Eng. 382 (2021) 113855. arXiv: 2012.00357, http://arxiv.org/abs/2012.00357, https://doi.org/10.1016/j.cma.2021.113855

[29] E. Prume, C. Gierden, M. Ortiz, S. Reese, et al., Direct data-driven algorithms for multiscale mechanics, Comp. Meth. Appl. Mech. Eng. 433 (2025) 117525. https://linkinghub.elsevier.com/retrieve/pii/S0045782524007795, https://doi.org/10.1016/j.cma.2024.117525

[30] L.T.K. Nguyen, R.C. Aydin, C.J. Cyron, Accelerating the distance-minimizing method for data-driven elasticity with adaptive hyperparameters, Comp. Mech. 70 (3) (2022) 621–638.

[31] Y. Kanno, Mixed-integer programming formulation of a data-driven solver in computational elasticity, Optim. Lett. 13 (7) (2019) 1505–1514. arXiv: 1810.04394, http://arxiv.org/abs/1810.04394, https://doi.org/10.1007/s11590-019-01409-w

[32] A. Leygue, M. Coret, J. Réthoré, L. Stainier, E. Verron, Data-based derivation of material response, Comp. Meth. Appl. Mech. Eng. 331 (2018) 184–196.

[33] A. Gorgogianni, K. Karapiperis, L. Stainier, M. Ortiz, J.E. Andrade, Adaptive goal-oriented data sampling in data-driven computational mechanics, Comp. Meth. Appl. Mech. Eng. 409 (2023) 115949.

[34] J.K. Cullum, R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. I: Theory, SIAM, 2002.

[35] R. Neal, MCMC using hamiltonian dynamics, in: S. Brooks, A. Gelman, G. Jones, X.-L. Meng (Eds.), Handbook of Markov Chain Monte Carlo, 20116022, Chapman and Hall/CRC, 2011. Series Title: Chapman & Hall/CRC Handbooks of Modern Statistical Methods, http://www.crcnetbase.com/doi/abs/10.1201/b10905-6, https://doi.org/10.1201/b10905-6

[36] A.M. Ferrenberg, D.P. Landau, R.H. Swendsen, Statistical errors in histogram reweighting, Phys. Rev. E 51 (5) (1995) 5092.

[37] L.T.K. Nguyen, M.-A. Keip, A data-driven approach to nonlinear elasticity, Comp. & Struct. 194 (2018) 97–115. https://linkinghub.elsevier.com/retrieve/pii/S0045794917301311, https://doi.org/10.1016/j.compstruc.2017.07.031