

Non-smooth optimization meets automated material model discovery

Moritz Flaschel^{a,b,*}, Trevor Hastie^c, Ellen Kuhl^{a,b}

^a Institute of Applied Mechanics, Egerlandstraße 5, Friedrich-Alexander-Universität Erlangen–Nürnberg, Erlangen, 91058, Germany

^b Department of Mechanical Engineering, Stanford University, 440 Escondido Mall, California, 94305, United States

^c Department of Statistics, Stanford University, Sequoia Hall 390 Jane Stanford Way, California, 94305-4065, United States

ARTICLE INFO

Keywords:

Non-smooth optimization

L_1 -norm regularization

LASSO

LARS

ISTA

Automated material model discovery

ABSTRACT

Automated material model discovery has gained significant traction in recent years, as it disrupts the tedious and time-consuming cycle of iteratively calibrating and modifying manually designed models. Non-smooth L_1 -norm regularization is the backbone of automated model discovery; however, the current literature on automated material model discovery offers limited insights into the robust and efficient minimization of non-smooth objective functions. In this work, we examine the minimization of functions of the form $f(\mathbf{w}) + \alpha \|\mathbf{w}\|_1$, where $\mathbf{w}$ are the material model parameters, f is a metric that quantifies the mismatch between the material model and the observed data, and $\alpha \geq 0$ is a regularization parameter that determines the sparsity of the solution. We investigate both the straightforward case where f is quadratic and the more complex scenario where it is non-quadratic or even non-convex. Importantly, in contrast to previous works on automated material model discovery, we do not only focus on methods that solve the sparse regression problem for a given value of the regularization parameter α , but propose methods to efficiently compute the entire regularization path, facilitating the selection of a suitable α . Specifically, we present four algorithms and discuss their roles for automated material model discovery in mechanics: First, we recapitulate a well-known *coordinate descent* algorithm that solves the minimization problem assuming that f is quadratic for a given value of α , also known as the LASSO. Second, we discuss the algorithm LARS, which automatically determines the critical values of α , at which material parameters in $\mathbf{w}$ are set to zero. Third, we propose to use the *proximal gradient method* ISTA for automated material model discovery if f is not quadratic, and fourth, we suggest a novel pathwise extension of ISTA for computing the regularization path. Many of these algorithms have not yet been applied to problems in computational material mechanics. We demonstrate the applicability of all algorithms for the automated discovery of incompressible hyperelastic material models from uniaxial tension and simple shear data.

1. Introduction

Traditional material modeling, that is, the manual design of a material model and the calibration of its material parameters, is known to be prone to modeling errors, for example, due to incorrect modeling assumptions or an inappropriate choice of functions describing material behavior. Consequently, current research focuses on machine learning material models [1–16], bypassing the

* Corresponding author.

E-mail address: moritz.flaschel@fau.de (M. Flaschel).

<https://doi.org/10.1016/j.jcp.2026.115043>

Received 25 September 2025; Received in revised form 29 January 2026; Accepted 12 May 2026

Available online 19 May 2026

0021-9991/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Table 1
Existing optimization methods for automated material model discovery based on L_1 - or L_p -regularization.

Method	References	Model is linear in the parameters $\boldsymbol{w}$	Compute regularization path
L-BFGS-B	[25]	Yes	No
Fixed-point iterations	[22]	Yes	No
Adam	[5]	No	No
Trust-region reflective algorithms	[38,58]	No	No

formulation of material models in the classical sense [17,18], or automatically discovering material models as interpretable symbolic expressions [19–35], see [36] for a comprehensive review.

While each data-driven material modeling method has its merits and is well suited to specific use cases, automatically discovering closed-form mathematical expressions for the material model offers several advantages. First, material models encoded in concise formulas are memory efficient because they compress all information about the material's behavior into short mathematical expressions with only a few parameters. Storing a handful of parameters requires less memory than storing the weights of a neural network or a database of stress-strain pairs, as needed for model-free approaches [17,18]. In addition, concise material models are typically more efficient to evaluate compared to other machine learning approaches in which information has to pass, for example, through several layers of a neural network. This means that concise models naturally lend themselves to finite element simulations [37]. Finally, automated material model discovery facilitates the physical interpretation of the discovered material behavior and simplifies the communication of the discovered models to other researchers. With automated material model discovery, we can identify the most suitable functions to describe the material behavior, determine the number of internal variables needed to capture its path-dependence, and automatically classify the material into an appropriate category, such as elasticity, viscoelasticity, or plasticity [38].

One of the most popular approaches to automated material model discovery are library-based approaches [22,25,28]. Given some data, these methods aim to select a material model from a large library of candidate models using sparse regression. The core idea of sparse regression is to add a sparsity-promoting L_1 -regularization term to the loss function that quantifies the mismatch between the model prediction and the data. By jointly minimizing the model-data mismatch and the regularization term, which is weighted by a regularization parameter α , this approach facilitates the discovery of concise material models that fit the data well. L_1 -regularization, or more generally L_p -regularization, first appeared in the context of model discovery in the early works of [39,40]. [41], who mathematically analyzed the L_1 -regularized problem, popularized the method under the name *Least Absolute Shrinkage and Selection Operator* (LASSO). Since then, L_1 -regularization has constituted the backbone of automated model discovery, and the concept has been mathematically analyzed and extended in various ways, for example, by [42–49]. Originally applied in statistics and data science, L_1 -regularization found its way into the physical sciences through the seminal work of [50], who proposed the method SINDy (*Sparse Identification of Nonlinear Dynamics*) to automatically discover short mathematical expressions for dynamic governing equations. This idea was adopted by the mechanics community to automatically discover material models from data using methods such as EUCLID (*Efficient Unsupervised Constitutive Law Identification and Discovery*) or CANNs (*Constitutive Artificial Neural Networks*), see, for example, [22,25,26,28,30,51–56] for hyperelastic materials, [57] for viscoelastic materials, [29,58,59] for plastic materials, [60] for an application to growth, and [38] for generalized standard materials.

A distinctive feature of the L_1 -regularization is its non-smoothness, which gives rise to a non-smooth optimization problem. While non-smooth optimization problems have been extensively studied within the mechanics community – particularly in the context of elasto-plasticity, see [61] and more recently [16,62,63] – they have not yet been rigorously examined in the context of material model discovery. Although L_1 -regularization is a well-established tool in the field of material model discovery, existing optimization strategies, such as fixed-point iterations [22], trust-region reflective algorithms [58], or gradient-based methods like the Adam optimizer [28] or L-BFGS-B [25], often do not fully exploit the structure of the underlying non-smooth optimization problem. Table 1 summarizes commonly used algorithms for mechanical material model discovery. Most of these methods do not explicitly leverage the existing body of knowledge in non-smooth optimization and therefore do not benefit from the theoretical results that have already been established in the non-smooth optimization community. In this work, we address this gap by investigating solvers that are specifically tailored for non-smooth L_1 -regularized optimization. This allows for a more principled and potentially more efficient approach to material model discovery, grounded in the theory of non-smooth optimization. We describe all algorithms, particularly the LARS algorithm and its internal mechanisms, in a level of detail specifically tailored to the mechanics community. For example, we discuss the *Coordinate Descent* (CD) or shooting algorithm [42,48,64] and the *Iterative Soft-Thresholding Algorithm* (ISTA) [65,66], which solve the L_1 -regularized problem for a given value of the regularization parameter α for material model libraries that are linear or nonlinear in the material parameters, respectively. In our numerical examples, we find that the CD algorithm is computationally faster than ISTA. However, ISTA is more generally applicable, as it imposes fewer constraints on the functional form of the objective function.

Importantly, existing methods for mechanical material model discovery focus on a given value of the regularization parameter, see Table 1. Here, we do not focus our attention solely on algorithms that solve the L_1 -regularized problem for a given value of α . Instead, we investigate algorithms for computing the entire regularization path, that is, the solutions of the L_1 -regularized problem for all possible values of the regularization parameter. Algorithms for computing the entire regularization path have not been studied in the context of mechanical material modeling before. One exception is the work of [67], which investigates LARS and related algorithms

Table 2

Overview of non-smooth optimization methods for automated material model discovery discussed in this contribution.

	Model is linear in parameters $\mathbf{w}$	Model is nonlinear in parameters $\mathbf{w}$
Given regularization parameter α	<i>Coordinate Descent</i> CD [42]	<i>Iterative Soft-Thresholding Algorithm</i> ISTA [65,66]
Compute regularization path	<i>Least Angle Regression</i> LARS-LASSO [43,45]	Pathwise ISTA (inspired by [48])

and was published shortly after we released our preprint. For material model libraries that depend linearly on the parameters, we discuss a modified version of *Least Angle Regression* LARS [43,45], which efficiently identifies the critical values of α at which changes in the material model are observed. Furthermore, inspired by the work of [48,68–70], we propose a pathwise extension of ISTA to compute the regularization path for material model libraries with nonlinear parameter dependencies. Table 2 summarizes the key solvers discussed in this work and highlights their specific use cases. For overviews of sparsity-promoting computational algorithms, we refer to the textbooks by [64,71] and the review articles by [72–74]. For related methods in the context of dimensionality reduction, we refer to [75]. Finally, for an overview in the context of polynomial chaos expansions, we refer to [76].

We note that – although being arguably the most prominent approach – L_1 -regularized regression does not constitute the only method for automated material model discovery. Alternative approaches include, for example, symbolic regression based on genetic algorithms [77–80]. These approaches have been used by [81] to discover system dynamics, and by [19–21,23,24,27,31,32,82] in the context of material modeling. Finally, neural-network-based approaches can be modified to yield interpretable expressions for the material model, as shown by [30,34,35].

Our paper is structured as follows. In Section 2, we present four mathematical problems that occur in the context of automated material model discovery. Subsequently, in Section 3, we discuss different solvers to approach the four mathematical problems. In Section 4, we place the mathematical problems in the context of material model discovery, and in Section 5, we apply the discussed methods to a set of benchmark problems.

2. Mathematical problems

The overarching goal of model discovery is to find mathematical models encoded in short mathematical expressions that are capable of describing a given dataset. Library-based approaches are one of the most popular strategies for model discovery [22,28,50]. The idea is to construct a general parametric ansatz for the model. This is also called the model library or model catalog, and it depends on a large number of parameters $\mathbf{w} \in \mathbb{R}^m$ with $m \gg 1$. The ability of the model to describe the given dataset is quantified by defining a metric $f(\mathbf{w})$, which measures the mismatch between the model prediction and the data. The specific form of $f(\mathbf{w})$ depends on the application and the availability of data. We will discuss different examples throughout this paper. Our primary objective is to find parameters $\mathbf{w}$ that minimize the model-data-mismatch

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}). \quad (1)$$

Due to the large number of parameters in the model library, it is capable of describing complex relationships in a given dataset. Thus, solving the above problem is likely to discover model parameters $\mathbf{w}^*$ for which the model describes the data well. However, describing the data well is not the only objective of model discovery. Our second objective is that the model is expressed by a concise mathematical formula, i.e., we seek to find models with a small number of nonzero parameters. The number of nonzero parameters in $\mathbf{w}$ is quantified by the L_0 -pseudo-norm

$$\|\mathbf{w}\|_0 = \sum_i I(w_i) \quad \text{with} \quad I(w_i) = \begin{cases} 1 & \text{if } w_i = 0 \\ 0 & \text{if } w_i \neq 0 \end{cases}, \quad (2)$$

which is not a proper norm because it violates the absolute homogeneity property. To limit the number of nonzero parameters, we can regularize the minimization problem in Eq. (1) by adding $\alpha \|\mathbf{w}\|_0$ with $\alpha \geq 0$ to the objective function

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}) + \alpha \|\mathbf{w}\|_0. \quad (3)$$

This regularization term penalizes solutions with many nonzero parameters and thus promotes sparsity in the solution vector. However, due to the discontinuous and non-convex regularization term, the problem in Eq. (3) is computationally expensive to solve for large numbers of parameters. In practice, the L_0 -pseudo-norm is often approximated by the p -th power of the L_p -pseudo-norm with $p > 0$ [22,40,83]

$$\|\mathbf{w}\|_p^p = \sum_i |w_i|^p. \quad (4)$$

The p -th power of the L_p -pseudo-norm is continuous and converges to the L_0 -pseudo-norm as p approaches zero, sharing the sparsity-promoting property of L_0 -regularization. The influence of the choice of p has been studied by [83]. In this work, we focus on the most common choice of $p = 1$ [41,50], which is the smallest value for which the L_p -pseudo-norm is convex and becomes a proper norm, thereby making the minimization problem easier to solve. We reformulate the problem in Eq. (3) as

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}) + \alpha \|\mathbf{w}\|_1 \quad \text{with} \quad \|\mathbf{w}\|_1 = \sum_i |w_i|. \quad (5)$$

The L_1 -regularization term was mathematically studied and popularized by [41,45,64,71], but it also appears in the early work by [39,40]. Due to the continuity and convexity of the L_1 -norm, the L_1 -regularized problem in Eq. (5) allows for an easier numerical treatment than the L_0 -regularized problem in Eq. (3), while retaining the sparsity-promoting property, as we will show at several occasions throughout this work. Similar to the general L_p -regularized problem, the L_1 -regularized problem yields solutions that range from fully dense at small values of α to completely sparse at larger values of α .

The problem posed in Eq. (5) raises two fundamental questions: First, noticing that the regularization term is non-smooth, how can the problem be solved efficiently and robustly using methods from the field of non-smooth optimization? And second, are there optimal strategies for choosing the regularization parameter α ? To answer these questions, we distinguish between four types of mathematical problems, which require different numerical solution strategies. Subsequently, we describe these problems and provide examples from the field of material model discovery.

2.1. Problem 1

In many practical applications, we are interested in models that depend linearly or affinely on the parameters $\mathbf{w} \in \mathbb{R}^m$ [22,40,41,50,57]. In the case of material modeling, models that depend linearly on the parameters appear, for example, in the constitutive theory of hyperelastic material models [22], as we will discuss in Section 4, or linear viscoelastic material models [57]. For such models, the model prediction $\boldsymbol{\mu} \in \mathbb{R}^n$ is assumed to be equal to a feature matrix $\mathbf{X}$ times the parameters $\mathbf{w}$, i.e., $\boldsymbol{\mu} = \mathbf{X}\mathbf{w}$. We denote the columns of the feature matrix as the feature vectors $\mathbf{X}_i$ and observe that the predictions are a linear combination of the feature vectors $\boldsymbol{\mu} = \mathbf{X}_1 w_1 + \dots + \mathbf{X}_m w_m$. We will assume throughout this work that the feature vectors $\mathbf{X}_i$ are linearly independent.

The most common choice for the model-data-mismatch is the sum of squares of the differences between the model prediction and the data

$$f(\mathbf{w}) = \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2. \quad (6)$$

As explained previously, we are interested in minimizing the model-data-mismatch, while at the same time penalizing the L_1 -norm of the parameters, see Eq. (5). Consequently, Problem 2.1 depicts the first type of problem considered in this work, in which we aim to minimize the L_1 -regularized model-data-mismatch for a given value of α . Problem 2.1 is also called the *Least Absolute Shrinkage and Selection Operator* (LASSO) [41].

Problem 1

Given $\mathbf{X} \in \mathbb{R}^{n \times m}$ with $\|\mathbf{X}_i\|_2 = 1$ and $\mathbf{y} \in \mathbb{R}^n$, define $f(\mathbf{w}) = \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$. For a given value of $\alpha \geq 0$, solve

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}) + \alpha \|\mathbf{w}\|_1. \quad (7)$$

We note that in Problem 2.1, we assume that the feature vectors are normalized, i.e., $\|\mathbf{X}_i\|_2 = 1$. This does not affect the generality of the problem. If, for example, a model with non-normalized feature vectors is given, e.g., $\boldsymbol{\mu} = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$, we can normalize the feature vectors according to $\mathbf{X}_i = \tilde{\mathbf{X}}_i / \|\tilde{\mathbf{X}}_i\|_2$, while scaling the associated parameters as $\tilde{\mathbf{w}}_i = \mathbf{w}_i / \|\tilde{\mathbf{X}}_i\|_2$. The prediction of the model is not affected because $\boldsymbol{\mu} = \tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{X}\mathbf{w}$. Choosing a large regularization parameter α in Problem 2.1 yields the zero solution $\mathbf{w}^* = \mathbf{0}$. As discussed in Appendix A, we are only interested in practically meaningful values of α that yield non-vanishing solutions. Finally, note that the regularized problem in Problem 2.1 is mathematically equivalent to the constrained problem

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \leq t, \quad (8)$$

where there is a one-to-one relationship between $\alpha > 0$ in Problem 2.1 and $t > 0$ in Eq. (8) [41,64,71].

2.2. Problem 2

As can be seen from Problem 2.1, the solution of the L_1 -regularized regression problem is dependent on the regularization parameter α . Thus, we may write the solution of the parameters as a function of the regularization parameter $\mathbf{w}^*(\alpha)$. Fig. 1a shows solutions to Problem 2.1 for different values of α for an exemplary dataset with five features. The dependence of the parameters on the choice of the regularization parameter is also referred to as the regularization path, or LASSO path for Problem 2.1 specifically. Because $\mathbf{X}$ and $\mathbf{y}$ typically originate from noisy measurements, it is likely that all parameters are nonzero if the problem is not regularized. This means that $\|\mathbf{w}^*(\alpha)\|_0 = m$ if $\alpha = 0$. Upon increasing the regularization parameter, more and more parameters are forced to be zero by the regularization term. Interestingly, the regularization path of Problem 2.1 is piecewise linear [45,49]. And we observe that, for certain values of the regularization parameter, the slopes of the functions $w_i^*(\alpha)$ change, see Fig. 1. We will refer to these values as the knots of the regularization path. The knots are the only points on the regularization path at which the sparsity of the solution changes.

In general, the sparsity of the solution $\|\mathbf{w}^*(\alpha)\|_0$ is not monotone in α , see Fig. 1b. However, we can define a subset of knots α_c with $c = 0, \dots, m$ and $0 = \alpha_m < \alpha_{m-1} < \dots < \alpha_1 < \alpha_0$ such that $\|\mathbf{w}^*(\alpha_c)\|_0 = c$ and $\|\mathbf{w}^*(\alpha)\|_0 > c$ for all $\alpha < \alpha_c$. We will refer to these knots as the critical values of the regularization parameter. When discovering models, we are mainly interested in the critical values of the regularization parameter. Solutions between critical values are not of interest, because between the critical values α_c and α_{c+1} , there

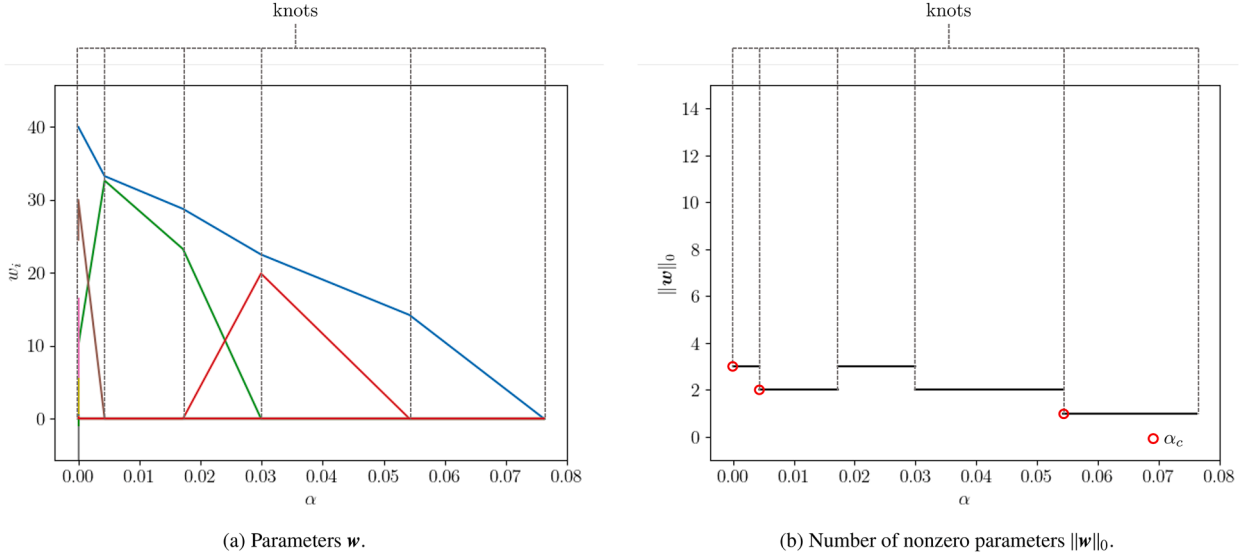


Fig. 1. Qualitative regularization path of Problem 2.1 for a representative dataset. At the knots, the slope of the piecewise linear regularization path changes. The critical values α_c , marked as red circles, are the lowest values of α required to achieve a specified sparsity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

exists no solution that at the same time has the same sparsity as the solution $w^*(\alpha_c)$ and exhibits a lower model-data-mismatch than $w^*(\alpha_c)$. This naturally leads to the question of whether we can directly determine the critical values α_c of the regularization parameter and the associated solution, as depicted in Problem 2.2.

Problem 2

Given $X \in \mathbb{R}^{n \times m}$ with $\|X_i\|_2 = 1$ and $y \in \mathbb{R}^n$, define $f(w) = \frac{1}{2n} \|y - Xw\|_2^2$. Given the problem

$$w^*(\alpha) = \arg \min_w f(w) + \alpha \|w\|_1, \quad (9)$$

find the critical values α_c with $c = 0, \dots, m$ and $0 = \alpha_m < \alpha_{m-1} < \dots < \alpha_1 < \alpha_0$ such that $\|w^*(\alpha_c)\|_0 = c$ and $\|w^*(\alpha)\|_0 > c$ for all $\alpha < \alpha_c$.^a

^aFor simplicity, we assume the existence of the sequence of values α_c .

We note that, in principle, it is possible that, for one critical value, two parameters become zero simultaneously. In other words, there may exist an α_c such that $\|w^*(\alpha_c)\|_0 = c$ and $\|w^*(\alpha)\|_0 \geq c + 2$ for all $\alpha < \alpha_c$. This is very unlikely for linearly independent feature vectors that originate from noisy measurements. In the following, we thus neglect this special case. Finally, we note that, in practice, the goal is typically not to identify all critical values as described in Problem 2.2, but rather to focus on the first few critical values that correspond to sparse solutions.

2.3. Problem 3

In Problems 2.1 and 2.2, we assumed models that depend linearly on the parameters w . In general, however, models may depend nonlinearly on their parameters. In the context of material modeling, this is, for example, the case for certain hyperelastic material models such as power-type Ogden models [54], exponential-type models [28], or dissipative material models [38,58]. For such models, the model prediction $\mu(w)$ is a nonlinear function of the parameters. The model-data-mismatch is therefore not quadratic and may even be non-convex. As the third type of problem that we consider in this work, see Problem 2.3, we consider the generalization of Problem 2.1, i.e., an L_1 -regularized problem in which the model-data-mismatch is not quadratic. We restrict our attention to models for which the model prediction $\mu(w)$ is differentiable with respect to the parameters such that also $f(w)$ is differentiable.

Problem 3

Given a differentiable function $f(w)$. For a given value of $\alpha \geq 0$, solve

$$w^* = \arg \min_w f(w) + \alpha \|w\|_1. \quad (10)$$

2.4. Problem 4

Problem 2.3 requires an a priori selection of the regularization parameter α . In practice, however, it is often not clear beforehand which value of α results in a model that is both sparse and demonstrates high fitting accuracy. Consequently, it can be beneficial to compute the regularization path for **Problem 2.3**. Unlike the regularization path visualized in Fig. 1, for models that depend nonlinearly on the parameters, the regularization path is not necessarily piecewise linear. For these models, identifying the critical values of α is not straightforward. A practical remedy is to solve **Problem 2.3** for a predefined set of values of α , as depicted in **Problem 2.4**. The values of α should be within a meaningful range, i.e., between zero and the smallest value of α , denoted by $\alpha^{(0)}$, that yields the zero parameter vector, the computation of which is discussed in **Appendix A**. As we discuss below, the computational burden of **Problem 2.4** is not simply n_α times the cost of **Problem 2.3**, since solutions at previous values of α can be used as initial estimates for subsequent computations, see [48,68–70].

Problem 4

Given a differentiable function $f(\mathbf{w})$, solve

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}) + \alpha^{(k)} \|\mathbf{w}\|_1, \quad (11)$$

for a predefined set of values $\alpha^{(k)}$ with $k = 0, \dots, n_\alpha - 1$ and $\alpha^{(k)} \in (0, \alpha^{(0)}]$, where $\alpha^{(0)} = \max_i \left| \frac{\partial f}{\partial w_i}(\mathbf{0}) \right|$.

3. Solution algorithms

3.1. Coordinate descent (CD)

A popular algorithm for solving **Problem 2.1** is the *coordinate descent* (CD) algorithm, also known as the shooting algorithm. It has been used and mathematically analyzed by [42,48].

We start with an initial guess for the parameters $\mathbf{w}^{(0)}$, for which we typically consider the ordinary least squares solution $\mathbf{w}^{(0)} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$, and iteratively compute a sequence $\mathbf{w}^{(k)}$ with $k = 0, 1, \dots$ that converges to the solution of **Problem 2.1**. The basic idea of the CD algorithm is to loop over all parameters at each step and treat all parameters except one as constants while minimizing the objective of **Problem 2.1**. Specifically, at each step k , we first set $\mathbf{w}^{(k)}$ to be equal to $\mathbf{w}^{(k-1)}$, and then for all $l = 1, \dots, m$ set $w_l^{(k)}$ to be equal to

$$w_l^{*(k)} = \arg \min_{w_l^{(k)}} \frac{1}{2n} \|\mathbf{X} \mathbf{w}^{(k)} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{w}^{(k)}\|_1. \quad (12)$$

Geometrically, each solving of Eq. (12) can be interpreted as minimizing along one coordinate $w_l^{(k)}$, while treating all other parameters as constants, which explains the name of the *coordinate descent* algorithm.

The strength of CD is that the minimization problem in Eq. (12) is a convex and non-smooth minimization problem that admits a closed-form and computationally efficient solution. In the following, we recapitulate the basic concepts of convex and non-smooth optimization and detail the solution of Eq. (12). First, we denote the objective function of Eq. (12) by $L(w_l^{(k)})$ and use index notation to rewrite it as

$$\begin{aligned} L(w_l^{(k)}) &= \frac{1}{2n} \sum_i \left[\sum_j X_{ij} w_j^{(k)} - y_i \right]^2 + \alpha \sum_j |w_j^{(k)}| \\ &= \underbrace{\frac{1}{2n} \sum_i \left[\sum_{j \neq l} X_{ij} w_j^{(k)} + X_{il} w_l^{(k)} - y_i \right]^2}_{L_f(w_l^{(k)})} + \underbrace{\alpha \sum_{j \neq l} |w_j^{(k)}| + \alpha |w_l^{(k)}|}_{L_\alpha(w_l^{(k)})}, \end{aligned} \quad (13)$$

where $w_j^{(k)}$ with $j \neq l$ are treated as constants. The first part of the objective function $L_f(w_l^{(k)})$ is smooth and differentiable. Its derivative computes to

$$\frac{\partial L_f}{\partial w_l^{(k)}} = \frac{1}{n} \sum_i X_{il} \left[\sum_{j \neq l} X_{ij} w_j^{(k)} + X_{il} w_l^{(k)} - y_i \right] = S_l^{(k)} + \frac{1}{n} \|\mathbf{X}_l\|_2^2 w_l^{(k)}, \quad (14)$$

where $S_l^{(k)} = \frac{1}{n} \sum_i X_{il} \left[\sum_{j \neq l} X_{ij} w_j^{(k)} - y_i \right]$. The second part of the objective function $L_\alpha(w_l^{(k)})$, however, is non-smooth and non-differentiable at $w_l^{(k)} = 0$, which consequently means that $L(w_l^{(k)})$ is non-smooth and non-differentiable.

From the theory of convex and non-smooth optimization [84,85], we recall the concepts of subderivatives and subdifferentials. A real number d is a subderivative of $L(w_l^{(k)})$ at the point $\hat{w}_l^{(k)}$ if $L(w_l^{(k)}) - L(\hat{w}_l^{(k)}) \geq d[w_l^{(k)} - \hat{w}_l^{(k)}]$ for all $w_l^{(k)}$. This can be interpreted as drawing a line of slope d through the point $\{\hat{w}_l^{(k)}, L(\hat{w}_l^{(k)})\}$. If the line is smaller than or equal to the graph of $L(w_l^{(k)})$ for all $w_l^{(k)}$,

then d is a subderivative. The set of all subderivatives of $L(w_l^{(k)})$ is called the subdifferential and will in the following be denoted by $\partial_{w_l^{(k)}} L$. A necessary and sufficient condition for a minimum of the convex function $L(w_l^{(k)})$ is that the subdifferential contains zero, $0 \in \partial_{w_l^{(k)}} L$, i.e., $d = 0$ is a subderivative at the minimum.

To derive the subdifferential of $L(w_l^{(k)})$, we consider $L_f(w_l^{(k)})$ and $L_\alpha(w_l^{(k)})$ separately. The first term, $L_f(w_l^{(k)})$, is smooth and differentiable, and its subderivative is uniquely determined by the derivative $\partial L_f / \partial w_l^{(k)}$ at each point. Thus, its subdifferential is the singleton $\partial_{w_l^{(k)}} L_f = \{\partial L_f / \partial w_l^{(k)}\}$. The second term, $L_\alpha(w_l^{(k)})$, is non-smooth and non-differentiable. Its subdifferential is identified as

$$\partial_{w_l^{(k)}} L_\alpha = \begin{cases} \{-\alpha\} & \text{if } w_l^{(k)} < 0 \\ [-\alpha, \alpha] & \text{if } w_l^{(k)} = 0 \\ \{\alpha\} & \text{if } w_l^{(k)} > 0 \end{cases}. \quad (15)$$

Finally, the subdifferential $\partial_{w_l^{(k)}} L$ is the sum of the subdifferentials $\partial_{w_l^{(k)}} L_f$ and $\partial_{w_l^{(k)}} L_\alpha$,

$$\partial_{w_l^{(k)}} L = \begin{cases} \{S_l^{(k)} + \frac{1}{n} \|X_l\|_2^2 w_l^{(k)} - \alpha\} & \text{if } w_l^{(k)} < 0 \\ [S_l^{(k)} - \alpha, S_l^{(k)} + \alpha] & \text{if } w_l^{(k)} = 0 \\ \{S_l^{(k)} + \frac{1}{n} \|X_l\|_2^2 w_l^{(k)} + \alpha\} & \text{if } w_l^{(k)} > 0 \end{cases}. \quad (16)$$

To solve the minimization problem in Eq. (12), we seek to find $w_l^{*(k)}$ such that $0 \in \partial_{w_l^{(k)}} L$. To this end, we consider the three cases:

- For $w_l^{*(k)} < 0$, it must be $S_l^{(k)} + \frac{1}{n} \|X_l\|_2^2 w_l^{*(k)} - \alpha = 0$, from which we deduce $w_l^{*(k)} = n(\alpha - S_l^{(k)}) / \|X_l\|_2^2$. This can only be an admissible solution if $w_l^{*(k)} < 0$ and thus if $S_l^{(k)} > \alpha$.
- For $w_l^{*(k)} = 0$, it must be $0 \in [S_l^{(k)} - \alpha, S_l^{(k)} + \alpha]$. This is only possible if $S_l^{(k)} - \alpha \leq 0$ and $S_l^{(k)} + \alpha \geq 0$, and thus if $-\alpha \leq S_l^{(k)} \leq \alpha$.
- For, $w_l^{*(k)} > 0$, analogously to the first case, we obtain $w_l^{*(k)} = n(-\alpha - S_l^{(k)}) / \|X_l\|_2^2$. This can only be an admissible solution if $w_l^{*(k)} > 0$ and thus if $S_l^{(k)} < -\alpha$.

Summarizing all three cases, we deduce the closed-form solution to Eq. (12)

$$w_l^{*(k)} = \begin{cases} n \frac{\alpha - S_l^{(k)}}{\|X_l\|_2^2} & \text{if } S_l^{(k)} > \alpha \\ 0 & \text{if } -\alpha \leq S_l^{(k)} \leq \alpha \\ n \frac{-\alpha - S_l^{(k)}}{\|X_l\|_2^2} & \text{if } S_l^{(k)} < -\alpha \end{cases}. \quad (17)$$

By introducing the so-called soft-thresholding function $\text{soft}_\alpha(x) = \text{sign}(x) \max\{|x| - \alpha, 0\}$, the closed-form solution is concisely written as

$$w_l^{*(k)} = -\frac{\text{soft}_\alpha(S_l^{(k)})}{\frac{1}{n} \|X_l\|_2^2}. \quad (18)$$

We observe that $w_l^{*(k)}$ is set exactly to zero if $-\alpha \leq S_l^{(k)} \leq \alpha$. Increasing $\alpha > 0$ increases the chance that $-\alpha \leq S_l^{(k)} \leq \alpha$ is fulfilled, and thus increases the sparsity of parameters. We finally notice that the closed-form solution further simplifies due to the normalization of X , i.e., $\|X_l\|_2^2 = 1$. The CD algorithm is summarized in Algorithm 1.

CD efficiently solves Problem 2.1 with proven convergence, see [42]. In principle, CD can also be used to numerically approach Problem 2.2, by simply solving Problem 2.1 for a large number of different values for α and determining approximations of the critical values α_c at which parameters are set to zero. However, such a brute force attempt to solving Problem 2.2 is computationally infeasible, especially for larger numbers of parameters. As we will discuss below, there exist more efficient algorithms to approach Problem 2.2. In special cases, CD can also be used to approach Problem 2.3. As proposed by [54], if the model is linearly dependent on many of the parameters and nonlinearly dependent on only a few of the parameters, the latter parameters can be discretized such that they vanish from the set of parameters to be determined in the optimization problem. Due to the discretization, Problem 2.3 is transformed to Problem 2.1 with a larger number of parameters, see [54] for details. However, this strategy is only applicable for a small number of nonlinear parameters. In the following, we will discuss algorithms beyond CD to efficiently solve Problems 2.2 and 2.3.

Dependent on the choice of the initial guess, CD can be considered as either a *top-down* or *bottom-up* approach. If the ordinary least squares solution $\mathbf{w}^{(0)} = [X^T X]^{-1} X^T \mathbf{y}$ is chosen as the initial guess, CD can be interpreted as a *top-down* approach. That is, starting with the ordinary least squares solution, which in general has many nonzero entries, the algorithm progressively sets more and more components of the parameter vector to zero during its iterations. If, however, $\mathbf{w}^{(0)} = \mathbf{0}$ is chosen as the initial guess, CD can be interpreted as a *bottom-up* approach that progressively adds more nonzero components to the parameter vector.

We note that CD can also be used to solve Problem 2.2 by solving Problem 2.1 for a predefined set of values α . Previously computed solutions for specific values of α can serve as initial guesses for neighboring values to improve efficiency. However, this strategy does not exploit the piecewise linear structure of the regularization path, as the LARS algorithm discussed below does.

Algorithm 1 *Coordinate descent (CD).*

Given $\mathbf{X}$ and $\mathbf{y}$
 Set initial guess $\mathbf{w}^{(0)} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$
 Choose the maximum number of steps NSTEP and convergence tolerance TOL
for $k = 1, \dots, \text{NSTEP}$ **do**
 $\mathbf{w}^{(k)} \leftarrow \mathbf{w}^{(k-1)}$
 for $l = 1, \dots, m$ **do**
 $S_l^{(k)} = \frac{1}{n} \sum_i X_{il} \left[\sum_{j \neq l} X_{ij} w_j^{(k)} - y_i \right]$
 $w_i^{(k)} \leftarrow \begin{cases} \frac{\alpha - S_l^{(k)}}{\frac{1}{n} \|\mathbf{X}_l\|_2^2} & \text{if } S_l^{(k)} > \alpha \\ 0 & \text{if } -\alpha \leq S_l^{(k)} \leq \alpha \\ \frac{-\alpha - S_l^{(k)}}{\frac{1}{n} \|\mathbf{X}_l\|_2^2} & \text{if } S_l^{(k)} < -\alpha \end{cases}$
 end for
 if $\|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|_2 < \text{TOL}$ **or** $|f(\mathbf{w}^{(k+1)}) + \alpha \|\mathbf{w}^{(k+1)}\|_1 - f(\mathbf{w}^{(k)}) - \alpha \|\mathbf{w}^{(k)}\|_1| < \text{TOL}$ **then**
 break
 end if
end for

3.2. Least angle regression (LARS and LARS-LASSO)

[45] proposed and mathematically analyzed the so-called *Least Angle Regression* (LARS), an algorithm for model selection that shares similarities with the earlier proposed *homotopy method* by [43]. This algorithm starts with the zero solution $\mathbf{w}^{(0)} = \mathbf{0}$ and builds a sequence of parameter vectors $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots$ by successively considering more nonzero parameters. In each step of the algorithm, only those parameters whose features show the highest absolute correlation with the residual $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{X}\mathbf{w}^{(k)}$ are modified. These parameters, which are called the active set, are updated toward the least squares solution of the problem until a new feature not belonging to the active set becomes equally correlated with the residual. Then, this feature is included in the active set and the process is repeated. As the algorithm progresses, the number of nonzero parameters increases and the model-data mismatch generally decreases. In this sense, LARS can be interpreted as an algorithm that belongs to the family of so-called *forward stepwise selection algorithms*, see [71]. Interestingly, [45] show that the solutions $\mathbf{w}^{(k)}$ obtained by LARS are similar to the solutions sought in Problem 2.2, and the authors propose a simple modification of LARS, which we call LARS-LASSO, to efficiently solve Problem 2.2. LARS-LASSO finds solutions to the LASSO Problem 2.1, but at the same time identifies the critical values of α at which the number of nonzero parameters changes. Thus, it provides an efficient tool to identify the LASSO regularization path. In the following, we explain LARS in detail for the special case of two features and the general case of m features, and show how to modify LARS to obtain LARS-LASSO.

3.2.1. LARS considering two features

Following [45], we focus on an example with two features and two parameters, i.e., $\mathbf{X} \in \mathbb{R}^{n \times 2}$ and $\mathbf{w} \in \mathbb{R}^2$, to explain the main ideas behind LARS and to visualize the algorithm.

Initially, all parameters are set to zero $\mathbf{w}^{(0)} = \mathbf{0}$, and the initial model prediction is computed as $\boldsymbol{\mu}^{(0)} = \mathbf{X}\mathbf{w}^{(0)} = \mathbf{0}$. At step k , the residual vector between the data $\mathbf{y}$ and the prediction $\boldsymbol{\mu}^{(k)}$ is defined as $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{X}\mathbf{w}^{(k)}$. The data can be additively divided into two parts $\mathbf{y} = \mathbf{y}_{\parallel} + \mathbf{y}_{\perp}$, such that the first part $\mathbf{y}_{\parallel}$ can be expressed as a linear combination of the linearly independent feature vectors $\mathbf{X}_1$ and $\mathbf{X}_2$, i.e., $\mathbf{y}_{\parallel} = \mathbf{X}_1 a_1 + \mathbf{X}_2 a_2$ with $a_1, a_2 \in \mathbb{R}$, while the second part $\mathbf{y}_{\perp}$ is perpendicular to the feature vectors, i.e., $\mathbf{X}_i^T \mathbf{y}_{\perp} = 0$. The first part $\mathbf{y}_{\parallel}$ is computed as the ordinary least squares solution $\mathbf{y}_{\parallel} = \mathbf{X} [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$, i.e., the projection of $\mathbf{y}$ onto the plane spanned by the feature vectors. Substituting $\mathbf{y} = \mathbf{y}_{\parallel} + \mathbf{y}_{\perp}$ into the residual, we obtain $\mathbf{r}^{(k)} = \mathbf{y}_{\parallel} - \mathbf{X}\mathbf{w}^{(k)} + \mathbf{y}_{\perp}$, and we define a parallel $\mathbf{r}_{\parallel}^{(k)} = \mathbf{y}_{\parallel} - \mathbf{X}\mathbf{w}^{(k)}$ and a perpendicular $\mathbf{r}_{\perp}^{(k)} = \mathbf{y}_{\perp}$ part of the residual. We notice that, no matter how we modify the parameters $\mathbf{w}^{(k)}$, the perpendicular part of the residual $\mathbf{r}_{\perp}^{(k)}$ remains unchanged.

Fig. 2 illustrates the first step of LARS for the example with two features. Recall that all parameters are initially zero $\mathbf{w}^{(0)} = \mathbf{0}$. In the first step of LARS, we set one of the parameters to be nonzero such that the prediction changes along the direction of the corresponding feature vector. To select which of the parameters should be set to nonzero, we seek to identify the feature vector that, upon adjusting its corresponding parameter, promises the greatest reduction of the norm of the residual at the next step $\mathbf{r}_{\parallel}^{(1)}$. To this end, we investigate the angles between the feature vectors $\mathbf{X}_i$ and the residual $\mathbf{r}_{\parallel}^{(0)}$

$$\beta_i^{(0)} = \angle(\mathbf{X}_i, \mathbf{r}_{\parallel}^{(0)}) = \cos^{-1} \frac{\mathbf{X}_i^T \mathbf{r}_{\parallel}^{(0)}}{\|\mathbf{X}_i\| \|\mathbf{r}_{\parallel}^{(0)}\|}. \quad (19)$$

Since we also allow for negative parameters, which means that the model prediction can also change along the direction of negative feature vectors, we are also interested in the angles between the negative feature vectors and the residual, i.e., $\beta_{-i}^{(0)} = \angle(-\mathbf{X}_i, \mathbf{r}_{\parallel}^{(0)})$.

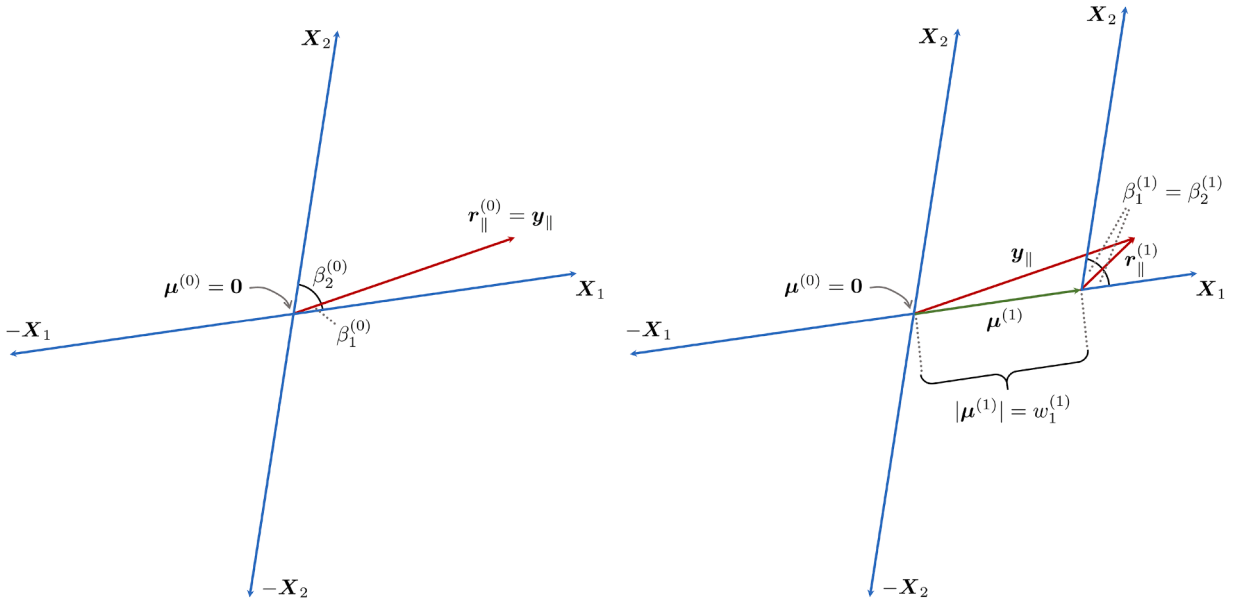


Fig. 2. Illustration of the first step of LARS considering two features. All vectors are illustrated in the two-dimensional plane spanned by the feature vectors X_1 and X_2 . The left figure illustrates the initial prediction $\mu^{(0)} = \mathbf{0}$ and the initial residual $r_{\parallel}^{(0)} = y_{\parallel}$. The feature vector X_1 shows the least angle to the residual $r_{\parallel}^{(0)}$. Therefore, we assume that, at the next step, the parameter $w_1^{(1)} \neq 0$ is nonzero, such that the prediction at the next step $\mu^{(1)}$ is parallel to X_1 . As illustrated in the right figure, $w_1^{(1)}$ is chosen such that the new residual $r_{\parallel}^{(1)}$ shows the same angle to both feature vectors.

Considering two features, this amounts to a total of four angles $\beta_1^{(0)}, \beta_2^{(0)}, \beta_{-1}^{(0)} = \pi - \beta_1^{(0)}, \beta_{-2}^{(0)} = \pi - \beta_2^{(0)}$. The feature vector or negative feature vector with the least angle to the residual promises the greatest reduction of the norm of the residual upon adjusting its corresponding parameter. This is the reason why the method is called *Least Angle Regression*.

It is not necessary to explicitly compute the angles. Instead, we compute the correlation vector $c^{(0)} = X^T r_{\parallel}^{(0)}$ with $c_i^{(0)} = X_i^T r_{\parallel}^{(0)}$. Because the feature vectors are normalized $\|X_i\| = 1$ and the function $\cos^{-1}(\cdot)$ in Eq. (19) is strictly monotonically decreasing, we can deduce that $\beta_i^{(0)} < \beta_j^{(0)}$ if and only if $c_i^{(0)} > c_j^{(0)}$. Further, it is $c_{-i}^{(0)} = -X_i^T r_{\parallel}^{(0)} = -c_i^{(0)}$. Thus, to identify the feature vector or negative feature vector corresponding to the smallest angle, it is sufficient to identify the feature vector with the greatest absolute correlation $\max_i |c_i^{(0)}|$, where the sign of the correlation $s_i^{(0)} = \text{sign}(c_i^{(0)})$ indicates whether the feature vector or its negative corresponds to the smallest angle.

After having identified the feature vector X_{i^*} with the greatest absolute correlation, i.e., $i^* = \arg \max_{i \in \{1,2\}} |c_i^{(0)}|$, we adjust the corresponding parameter $w_{i^*}^{(1)} = w_{i^*}^{(0)} + \Delta w^{(0)}$. The prediction at the next step is therefore $\mu^{(1)} = \mu^{(0)} + X_{i^*} \Delta w^{(0)} = X_{i^*} \Delta w^{(0)}$ and the residual is $r_{\parallel}^{(1)} = y_{\parallel} - X_{i^*} \Delta w^{(0)}$. The adjustment $\Delta w^{(0)}$ could, for example, be chosen such that $\|r_{\parallel}^{(1)}\|^2$ is minimized, which would result in $\Delta w^{(0)} = (X_{i^*}^T X_{i^*})^{-1} X_{i^*}^T y_{\parallel} = c_{i^*}^{(0)}$, where we made use of the fact that the feature vectors are normalized. Choosing $\Delta w^{(0)}$ in this way is the core idea of so-called *forward stepwise selection methods*, see the overview by [71]. However, *forward stepwise selection methods* are known to disregard features from the active set, even when they exhibit a strong correlation with the data. As a result, [45] describe them as "overly greedy". On the contrary, the idea of LARS is to adjust $\Delta w^{(0)}$ until another feature vector X_{j^*} with $i^* \neq j^*$ becomes equally correlated in absolute value with the residual. This means that we seek to choose $\Delta w^{(0)}$ such that $|c_{i^*}^{(1)}| = |c_{j^*}^{(1)}|$. For the example with two features, this means that either $c_{i^*}^{(1)} = c_{j^*}^{(1)}$ or $c_{i^*}^{(1)} = -c_{j^*}^{(1)}$. The first condition leads to

$$\begin{aligned} c_{i^*}^{(1)} = c_{j^*}^{(1)} &\Rightarrow X_{i^*}^T r_{\parallel}^{(1)} = X_{j^*}^T r_{\parallel}^{(1)} \\ &\Rightarrow X_{i^*}^T (y_{\parallel} - X_{i^*} \Delta w^{(0)}) = X_{j^*}^T (y_{\parallel} - X_{i^*} \Delta w^{(0)}) \\ &\Rightarrow \Delta w^{(0)} = \frac{X_{i^*}^T y_{\parallel} - X_{j^*}^T y_{\parallel}}{X_{i^*}^T X_{i^*} - X_{j^*}^T X_{i^*}} = \frac{c_{i^*}^{(0)} - c_{j^*}^{(0)}}{1 - X_{j^*}^T X_{i^*}}, \end{aligned} \quad (20)$$

while the second condition leads to

$$c_{i^*}^{(1)} = -c_{j^*}^{(1)} \Rightarrow \Delta w^{(0)} = \frac{X_{i^*}^T y_{\parallel} + X_{j^*}^T y_{\parallel}}{X_{i^*}^T X_{i^*} + X_{j^*}^T X_{i^*}} = \frac{c_{i^*}^{(0)} + c_{j^*}^{(0)}}{1 + X_{j^*}^T X_{i^*}}. \quad (21)$$

In Eqs. (20) and (21), we made use of the normalization of the feature vectors $X_{i^*}^T X_{i^*} = 1$. Further, due to the normalization, it is $-1 < X_{j^*}^T X_{i^*} < 1$. This means that the denominator in Eqs. (20) and (21) is positive. Because of $|c_{i^*}^{(0)}| > |c_{j^*}^{(0)}|$, the sign of the numerator

is $\text{sign}(c_{i^*}^{(0)})$. Therefore, we can deduce that $\text{sign}(\Delta w^{(0)}) = \text{sign}(c_{i^*}^{(0)})$, which means that the w_{i^*} is updated such that it shares the sign with the correlation of the corresponding feature vector.

Among the two potential values for $\Delta w^{(0)}$ in Eqs. (20) and (21), we choose the one with the smallest absolute value. After computing $\Delta w^{(0)}$ and thus $w^{(1)}$, which has one nonzero entry, we finally compute the solution vector with all parameters being nonzero as the ordinary least squares solution $w^{(2)} = [X^T X]^{-1} X^T y$.

3.2.2. LARS for more than two features

We now consider LARS [45] for the general case with more than two features, i.e., $X \in \mathbb{R}^{n \times m}$ and $w \in \mathbb{R}^m$. As for the case with two features, LARS starts by setting all parameters to zero $w^{(0)} = 0$ and builds a sequence of parameter vectors $w^{(0)}, w^{(1)}, \dots$ by successively considering more nonzero parameters. From Section 3.2.1, we recapitulate the projection of y onto the plane spanned by the feature vectors $y_{\parallel} = X [X^T X]^{-1} X^T y$, the prediction $\mu^{(k)} = X w^{(k)}$, the parallel part of the residual $r_{\parallel}^{(k)} = y_{\parallel} - \mu^{(k)}$, and the correlations between each feature and the residual $c^{(k)} = X^T r_{\parallel}^{(k)}$ with $c_i^{(k)} = X_i^T r_{\parallel}^{(k)}$.

At each step of LARS, we identify those feature vectors that exhibit the greatest absolute correlation with the residuals. Specifically, we define the active set and its complement

$$\mathcal{A}^{(k)} = \left\{ i^* \in \{1, \dots, m\} \mid |c_{i^*}^{(k)}| = \bar{c}_{\max}^{(k)} = \max_i |c_i^{(k)}| \right\}, \quad \mathcal{A}^{\complement(k)} = \left\{ j^* \in \{1, \dots, m\} \mid j^* \notin \mathcal{A}^{(k)} \right\}. \quad (22)$$

In the following, we will use the indices i^* and j^* to refer to elements in the active set and its complement at the current step k , respectively.

The signs of the correlations indicate whether the corresponding feature vector or its negative counterpart exhibits a greater correlation with the residual. Thus, the signs indicate whether the parameters corresponding to the feature vectors should increase or decrease in order to reduce the residual at the next step. We define the vector $s^{(k)}$ such that $s_i^{(k)} = \text{sign}(c_i^{(k)})$, and we flip the signs of the feature vectors $\bar{X}_i^{(k)} = s_i^{(k)} X_i$, such that their correlations with the residual are positive $\bar{c}_i^{(k)} = \bar{X}_i^{(k)T} r_{\parallel}^{(k)} = s_i^{(k)} c_i^{(k)} = |c_i^{(k)}|$.

LARS updates the prediction at each step according to the rule

$$\mu^{(k+1)} = \mu^{(k)} + \gamma^{(k)} u^{(k)}, \quad (23)$$

where the unit vector $u^{(k)}$ with $\|u^{(k)}\| = 1$ defines the direction of the update and $\gamma^{(k)} > 0$ defines the step size. This consequently means that the correlations are updated according to

$$c_i^{(k+1)} = X_i^T r_{\parallel}^{(k+1)} = X_i^T (y_{\parallel} - \mu^{(k)} - \gamma^{(k)} u^{(k)}) = c_i^{(k)} - \gamma^{(k)} a_i^{(k)}, \quad (24)$$

where we defined $a_i^{(k)} = X_i^T u^{(k)}$. In the following, we detail the choice of $u^{(k)}$ and $\gamma^{(k)}$.

At the current step, all active feature vectors $\bar{X}_{i^*}^{(k)}$ exhibit the same absolute correlation with the residual, i.e., $\bar{c}_{i^*}^{(k)}$ is the same for all $i^* \in \mathcal{A}^{(k)}$. The idea of LARS is to choose the unit vector $u^{(k)}$ such that the active feature vectors also exhibit the same absolute correlation with the residual at the next step. Note that we will see later in this section that $\text{sign}(c_{i^*}^{(k+1)}) = \text{sign}(c_{i^*}^{(k)})$ and thus $\bar{X}_{i^*}^{(k+1)} = \bar{X}_{i^*}^{(k)}$. Consequently, the absolute correlations at the next step are $\bar{c}_{i^*}^{(k+1)} = \bar{X}_{i^*}^{(k)T} r_{\parallel}^{(k+1)} = \bar{c}_{i^*}^{(k)} - \gamma^{(k)} \bar{a}_{i^*}^{(k)}$, with $\bar{a}_{i^*}^{(k)} = \bar{X}_{i^*}^{(k)T} u^{(k)}$. To achieve that the active feature vectors also exhibit the same absolute correlation with the residual at the next step, $\bar{a}_{i^*}^{(k)}$ must be equal for all $i^* \in \mathcal{A}^{(k)}$, i.e., the unit vector $u^{(k)}$ must be chosen such that it is equiangular to all active feature vectors $\bar{X}_{i^*}^{(k)}$. The choice of the equiangular vector is not unique. We are specifically interested in the equiangular vector that is in the span of $\bar{X}_{i^*}^{(k)}$ and has a positive correlation with $\bar{X}_{i^*}^{(k)}$. To compute the equiangular vector $u^{(k)}$, we define the matrix $\bar{X}_{\mathcal{A}}^{(k)}$ which is composed of the columns $\bar{X}_{i^*}^{(k)}$ for all $i^* \in \mathcal{A}^{(k)}$ and we define the vector of ones $\mathbf{1}_{\mathcal{A}}$. The equiangular vector must fulfill the relationship $\bar{X}_{\mathcal{A}}^{(k)T} u^{(k)} = a \mathbf{1}_{\mathcal{A}}$ where a is a scalar. By multiplying $\mathbf{1}_{\mathcal{A}}$ with $\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}$ and its inverse, we obtain $\mathbf{1}_{\mathcal{A}} = \bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)} [\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$. Thus, $\bar{X}_{\mathcal{A}}^{(k)} [\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$ is an equiangular vector for $a = 1$, which we normalize to obtain

$$u^{(k)} = A^{(k)} \bar{X}_{\mathcal{A}}^{(k)} [\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}} \quad \text{with} \quad A^{(k)} = \frac{1}{\|\bar{X}_{\mathcal{A}}^{(k)} [\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}\|} = \frac{1}{\sqrt{\mathbf{1}_{\mathcal{A}}^T [\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}}} > 0. \quad (25)$$

The inverse in the formula above is not computed explicitly. Instead, $[\bar{X}_{\mathcal{A}}^{(k)T} \bar{X}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$ is computed by solving a linear system of equations. We notice that $\bar{X}_{\mathcal{A}}^{(k)T} u^{(k)} = A^{(k)} \mathbf{1}_{\mathcal{A}}$, which means that the correlations $\bar{X}_{i^*}^{(k)T} u^{(k)} = A^{(k)}$ are positive, such that the angles between $\bar{X}_{i^*}^{(k)}$ and $u^{(k)}$ are smaller than $\pi/2$.

After computing the direction $u^{(k)}$, we are left with identifying the step size $\gamma^{(k)}$. The step size is always chosen positive, and it is chosen such that a new feature X_{j^*} enters the active set at the next step, i.e., $j^* \in \mathcal{A}^{(k+1)}$, but $j^* \notin \mathcal{A}^{(k)}$. We are interested in the smallest possible positive step size, such that there exists an X_{j^*} with $|c_{j^*}^{(k+1)}| = \bar{c}_{\max}^{(k+1)}$, i.e., either $c_{j^*}^{(k+1)} = \bar{c}_{\max}^{(k+1)}$ or $-c_{j^*}^{(k+1)} = \bar{c}_{\max}^{(k+1)}$. The first condition leads to

$$c_{j^*}^{(k+1)} = \bar{c}_{\max}^{(k+1)} \Rightarrow c_{j^*}^{(k)} - \gamma^{(k)} a_{j^*}^{(k)} = \bar{c}_{\max}^{(k)} - \gamma^{(k)} A^{(k)} \Rightarrow \gamma^{(k)} = \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}}, \quad (26)$$

while the second condition leads to

$$-c_{j^*}^{(k+1)} = \bar{c}_{\max}^{(k+1)} \Rightarrow \gamma^{(k)} = \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}}. \quad (27)$$

Among all potential values for $\gamma^{(k)}$ in Eqs. (26) and (27), we choose the smallest positive value

$$\gamma^{(k)} = \min_{j^* \in \mathcal{A}^{(k)}} \left\{ \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}}, \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}} \right\}. \quad (28)$$

After having identified $\mathbf{u}^{(k)}$ and $\gamma^{(k)}$, the prediction at the next step of LARS is computed according to Eq. (23). Substituting $\mathbf{u}^{(k)}$ into Eq. (23) gives

$$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \gamma^{(k)} A^{(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}. \quad (29)$$

We split the prediction into the contributions from the inactive and active parameters $\boldsymbol{\mu}^{(k)} = \mathbf{X} \mathbf{w}^{(k)} = \mathbf{X}_{\mathcal{A}^c} \mathbf{w}_{\mathcal{A}^c}^{(k)} + \mathbf{X}_{\mathcal{A}} \mathbf{w}_{\mathcal{A}}^{(k)}$. Further, we use $\bar{\mathbf{X}}_{\mathcal{A}}^{(k)} = \mathbf{X}_{\mathcal{A}} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)})$ to arrive at

$$\begin{aligned} \boldsymbol{\mu}^{(k+1)} &= \mathbf{X}_{\mathcal{A}^c} \mathbf{w}_{\mathcal{A}^c}^{(k)} + \mathbf{X}_{\mathcal{A}} \mathbf{w}_{\mathcal{A}}^{(k)} + \gamma^{(k)} A^{(k)} \mathbf{X}_{\mathcal{A}} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}} \\ &= \mathbf{X}_{\mathcal{A}^c} \mathbf{w}_{\mathcal{A}^c}^{(k)} + \mathbf{X}_{\mathcal{A}} \underbrace{\left[\mathbf{w}_{\mathcal{A}}^{(k)} + \gamma^{(k)} A^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}} \right]}_{\Delta \mathbf{w}_{\mathcal{A}}^{(k)}}, \end{aligned} \quad (30)$$

where we identify the update of the active parameters $\Delta \mathbf{w}_{\mathcal{A}}^{(k)}$ in each step of LARS. See Algorithm 2 for a summary of the LARS algorithm.

Algorithm 2 Least angle regression (LAR).

Given $\mathbf{X}$ and $\mathbf{y}$

Set $k = 0$, $\mathbf{w}^{(0)} = \mathbf{0}$

$\mathbf{y}_{\parallel} = \mathbf{X} [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$

while $\|\mathbf{w}^{(k)}\|_0 < m - 1$ **do**

$\mathbf{c}^{(k)} = \mathbf{X}^T [\mathbf{y}_{\parallel} - \mathbf{X} \mathbf{w}^{(k)}]$

$\mathcal{A}^{(k)} = \left\{ i^* \in \{1, \dots, m\} \mid |c_{i^*}^{(k)}| = \bar{c}_{\max}^{(k)} = \max_i |c_i^{(k)}| \right\}$, $\mathcal{A}^{(k)} = \left\{ j^* \in \{1, \dots, m\} \mid j^* \notin \mathcal{A}^{(k)} \right\}$

$\mathbf{s}^{(k)} = \text{sign}(\mathbf{c}^{(k)})$

$\bar{\mathbf{X}}_{\mathcal{A}}^{(k)} = \mathbf{X}_{\mathcal{A}} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)})$

$A^{(k)} = 1 / \sqrt{\mathbf{1}_{\mathcal{A}}^T \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}}$

$\mathbf{u}^{(k)} = A^{(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}$

$\gamma^{(k)} = \min_{j^* \in \mathcal{A}^{(k)}}^+ \left\{ \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}}, \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}} \right\}$

$\mathbf{w}_{\mathcal{A}^c}^{(k+1)} \leftarrow \mathbf{w}_{\mathcal{A}^c}^{(k)}$

$\mathbf{w}_{\mathcal{A}}^{(k+1)} \leftarrow \mathbf{w}_{\mathcal{A}}^{(k)} + \gamma^{(k)} A^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}$

$k \leftarrow k + 1$

end while

$\mathbf{w}^{(k+1)} \leftarrow [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$

Choosing the step size as shown in Eq. (B.1) has an interesting effect on the evolution of correlations corresponding to the active set over each step of LARS. Specifically, we recall that $c_{i^*}^{(k+1)} = c_{i^*}^{(k)} - \gamma^{(k)} \mathbf{X}_{i^*}^T \mathbf{u}^{(k)} = \text{sign}(c_{i^*}^{(k)}) [\bar{c}_{\max}^{(k)} - \gamma^{(k)} A^{(k)}]$. From Eq. (B.1), we deduce that $\gamma^{(k)} A^{(k)} < \bar{c}_{\max}^{(k)}$; see Appendix B.1, which means that the signs of the correlations corresponding to the active set do not change over one step, $\text{sign}(c_{i^*}^{(k+1)}) = \text{sign}(c_{i^*}^{(k)})$. And at each step, the maximum absolute correlation decreases according to $|c_{i^*}^{(k+1)}| = \bar{c}_{\max}^{(k)} - \gamma^{(k)} A^{(k)}$.

3.2.3. LARS-LASSO

The previously described LARS algorithm provides an efficient tool for model discovery. The solutions $\mathbf{w}^{(k)}$ computed by LARS are similar to those obtained by solving Problem 2.1 for different values of α . However, under certain conditions, LARS can yield solutions that cannot be identified as solutions of Problem 2.1. Specifically, it can be shown [45] that any nonzero parameter $w_i^* \neq 0$ of a solution to Problem 2.1 must fulfill $\text{sign}(w_i^*) = \text{sign}(c_i)$ with $c_i = \mathbf{X}_i^T [\mathbf{y}_{\parallel} - \mathbf{X} \mathbf{w}^*]$, and LARS can yield solutions that violate this

condition. Thus, [45] proposed a modification of LARS, which we call LARS-LASSO, that is specifically designed to find solutions to Problem 2.1.

The algorithm starts off by initializing all parameters to zero $\mathbf{w}^{(0)} = \mathbf{0}$. This is a valid solution to Problem 2.1 for a sufficiently large value of α , as discussed in Appendix A. At each step, we assume that $\text{sign}(w_i^{(k)}) = \text{sign}(c_i^{(k)})$ for $w_i^{(k)} \neq 0$, and we modify the LARS steps, such that, after each step, the condition $\text{sign}(w_i^{(k+1)}) = \text{sign}(c_i^{(k+1)})$ for $w_i^{(k+1)} \neq 0$ is fulfilled. As discussed after Eq. (25), the signs of the correlations corresponding to the active set do not change over one step. We distinguish two scenarios: First, we consider the case where $w_i^{(k)} = 0$ and $w_i^{(k+1)} \neq 0$. For this case, it is $\text{sign}(w_i^{(k+1)}) = \text{sign}(c_i^{(k)})$, see Appendix B.2, and therefore $\text{sign}(w_i^{(k+1)}) = \text{sign}(c_i^{(k+1)})$, which means that the condition is satisfied. Second, the condition may be violated if $w_i^{(k)} \neq 0$ and $\text{sign}(w_i^{(k+1)}) \neq \text{sign}(w_i^{(k)})$. We recall the update of the active parameters in each step, see Eq. (30),

$$\mathbf{w}_{\mathcal{A}}^{(k+1)} = \mathbf{w}_{\mathcal{A}}^{(k)} + \Delta \mathbf{w}_{\mathcal{A}}^{(k)} = \mathbf{w}_{\mathcal{A}}^{(k)} + \gamma^{(k)} \mathbf{A}^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}, \quad (31)$$

which we write in index notation as

$$w_{i^*}^{(k+1)} = w_{i^*}^{(k)} + \Delta w_{i^*}^{(k)} = w_{i^*}^{(k)} + \gamma^{(k)} d_{i^*}^{(k)}, \quad (32)$$

where $i^* \in \mathcal{A}^{(k)}$ and $d_{i^*}^{(k)}$ are the associated entries of the vector $\mathbf{A}^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}$. We recall that $\gamma^{(k)} > 0$ and observe that the sign of the parameters changes if and only if $\text{sign}(w_{i^*}^{(k)}) \neq \text{sign}(d_{i^*}^{(k)})$ and $\gamma^{(k)} > -w_{i^*}^{(k)} / d_{i^*}^{(k)}$ for any $i^* \in \mathcal{A}^{(k)}$. We define $\tilde{\gamma} = \min_{i^* \in \mathcal{A}^{(k)}} \{-w_{i^*}^{(k)} / d_{i^*}^{(k)}\}$ as the smallest possible positive value of $\gamma^{(k)}$ for which one of the parameters switches its sign. For the special case that $\text{sign}(w_{i^*}^{(k)}) = \text{sign}(d_{i^*}^{(k)})$ for all $i^* \in \mathcal{A}$, we set $\tilde{\gamma} = \infty$. We finally modify the update rule such that, if the stepsize $\gamma^{(k)}$ computed by (B.1) is greater than $\tilde{\gamma}$, we set instead $\gamma^{(k)} = \tilde{\gamma}$. In this way, the parameter $w_{\tilde{i}}^{(k)}$ with $\tilde{i} \in \mathcal{A}^{(k)}$ that would first switch its sign upon increasing $\gamma^{(k)}$ equates to zero after the step $w_{\tilde{i}}^{(k+1)} = 0$. Note that we follow [45] and assume that for $\gamma^{(k)} = \tilde{\gamma}$, only one parameter equates to zero. The index $\tilde{i}$ that corresponds to this parameter is excluded from the active set at the next step.

Each solution $\mathbf{w}^{(k)}$ obtained by LARS-LASSO is a solution to Problem 2.1 [45]. Given the solution $\mathbf{w}^{(k)}$, the corresponding regularization parameter $\alpha^{(k)}$ can be computed as detailed in Appendix B.3. Specifically, we obtain

$$\alpha^{(k)} = \frac{\tilde{c}_{\max}^{(k)}}{n}. \quad (33)$$

The modified algorithm LARS-LASSO is summarized in Algorithm 5 in Appendix C.

LARS-LASSO can be interpreted as a *bottom-up* approach, as it starts from the zero solution and successively adds nonzero components to the parameter vector. As discussed in Section 5, when discovering models to describe the mechanical behavior of materials, we are typically interested in models with a small number of material parameters, often as few as two or three for isotropic materials. This makes *bottom-up* approaches more favorable in practice, as they are expected to identify the practically relevant models more efficiently than *top-down* methods.

LARS-LASSO efficiently solves Problem 2.2, however, we note that LARS-LASSO may also be used to efficiently solve Problem 2.1. If the critical values of the regularization parameters and the corresponding parameters $\mathbf{w}^*(\alpha_c)$ are known, see Problem 2.2, solutions for regularization parameters between critical values can be obtained through linear interpolation between the parameters $\mathbf{w}^*(\alpha_c)$. LARS-LASSO is computationally more efficient than CD, especially if we are only interested in the large regularization parameter regime, i.e., the first steps of LARS-LASSO.

3.3. Iterative soft-thresholding algorithm (ISTA)

We finally move on to models that depend nonlinearly on the parameters and discuss strategies for numerically solving Problem 2.3. Specifically, we put our attention on the *Iterative Soft-Thresholding Algorithm* (ISTA) which is a first-order method belonging to the family of *proximal gradient methods* [65,66]. ISTA is shown in Algorithm 3 and briefly described in the following. For a detailed description and a mathematical treatment of the method, we refer to [66].

Algorithm 3 Iterative soft-thresholding algorithm (ISTA).

Given f
 Set initial guess $\mathbf{w}^{(0)}$
 Choose the maximum number of steps NSTEP and convergence tolerance TOL
for $k = 0, \dots, \text{NSTEP} - 1$ **do**
 $\mathbf{w}^{(k+1)} = \text{prox}_{\gamma g}(\mathbf{w}^{(k)} - \gamma \nabla f(\mathbf{w}^{(k)}))$
 if $\|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|_2 < \text{TOL}$ **or** $|f(\mathbf{w}^{(k+1)}) + g(\mathbf{w}^{(k+1)}) - f(\mathbf{w}^{(k)}) - g(\mathbf{w}^{(k)})| < \text{TOL}$ **then**
 break
 end if
end for

Problem 2.3 can be mathematically interpreted as a so-called composite problem [66], a multi-objective optimization problem, or, more simply, a regularized optimization problem

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} [f(\mathbf{w}) + g(\mathbf{w})], \quad (34)$$

where for our case $g(\mathbf{w}) = \alpha \|\mathbf{w}\|_1$. *Proximal gradient methods* constitute a family of algorithms designed to solve composite problems. They are first order methods relying on gradient computations of $f(\mathbf{w})$ and thus share similarities with classical gradient descent algorithms. They start off with an initial guess $\mathbf{w}^{(0)}$ and evaluate the gradient $\nabla f(\mathbf{w}^{(k)}) = \partial f / \partial \mathbf{w}(\mathbf{w}^{(k)})$ at each iteration k . As in classical gradient descent algorithms, the current solution is updated by making a step of a given step size $\gamma > 0$ into the direction of the negative gradient, i.e., $\mathbf{w}^{(k)} - \gamma \nabla f(\mathbf{w}^{(k)})$. Afterwards, and in contrast to classical gradient descent algorithms, a so-called proximal mapping $\text{prox}_{\gamma g}(\cdot)$ is applied, such that one step of the *proximal gradient methods* can be summarized as

$$\mathbf{w}^{(k+1)} = \text{prox}_{\gamma g}(\mathbf{w}^{(k)} - \gamma \nabla f(\mathbf{w}^{(k)})). \quad (35)$$

The proximal mapping or proximal operator of a function $h(\mathbf{w})$ is defined through

$$\text{prox}_h(\mathbf{w}) = \arg \min_{\mathbf{u}} \left[h(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|_2^2 \right]. \quad (36)$$

Thus, the proximal mapping $\text{prox}_{\gamma g}(\mathbf{w}^{(k)})$ in Eq. (35) with $g(\mathbf{w}) = \alpha \|\mathbf{w}\|_1$ is

$$\text{prox}_{\gamma g}(\mathbf{w}) = \arg \min_{\mathbf{u}} \left[\gamma \alpha \|\mathbf{u}\|_1 + \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|_2^2 \right]. \quad (37)$$

We notice the similarity between the minimization problem above and **Problem 2.1** with the feature matrix being the identity $\mathbf{X} = \mathbf{I}$. This problem has a closed-form solution, which is written in index notation as [66]

$$\{\text{prox}_{\gamma g}(\mathbf{w})\}_i = \text{soft}_{\gamma \alpha}(w_i) = \text{sign}(w_i) \max\{|w_i| - \gamma \alpha, 0\} = \begin{cases} w_i - \gamma \alpha & \text{if } w_i > \gamma \alpha \\ 0 & \text{if } -\gamma \alpha \leq w_i \leq \gamma \alpha \\ w_i + \gamma \alpha & \text{if } w_i < -\gamma \alpha \end{cases}, \quad (38)$$

which we identify as the soft-thresholding function similar to Eqs. (17) and (18).

The convergence proof of ISTA for solving **Problem 2.3** with $\nabla f(\mathbf{w})$ being Lipschitz continuous can be found in [66]. We note that the choice of the step size directly influences the convergence behavior of ISTA. If the step size is chosen too large, ISTA does not converge, and if the step size is chosen too small, the number of iterations to achieve convergence increases. If the Lipschitz constant of $\nabla f(\mathbf{w})$ is known, the step size can be chosen dependent on the Lipschitz constant [66]. In practice, however, the Lipschitz constant is usually not known such that a suitable step size must be determined through trial and error. Obviously, ISTA can also be applied to **Problem 2.1**. However, CD typically outperforms ISTA in terms of computational efficiency for **Problem 2.1**. Further, CD is preferred over ISTA because it does not require choosing a step size.

We note that, dependent on the initial guess, ISTA can be interpreted as either a *bottom-up* approach or a *top-down* approach. If the zero vector is used as the initial guess, ISTA successively adds more nonzero parameters to the solution. In contrast, if a dense vector is used as the initial guess, ISTA progressively sets more and more parameters to zero.

3.4. Pathwise ISTA

ISTA solves **Problem 2.3** for a given value of α . However, to select a suitable value of α , it is often beneficial to compute the regularization path of **Problem 2.3**. Computing the regularization path for models that depend nonlinearly on the parameters is not straightforward. Yet, in the literature, different methods for computing approximations of the regularization path have been proposed [48,68–70]. These methods start off with a value of α that yields the zero solution and then successively decrease α in predefined step sizes to approximately compute the regularization path, see **Problem 2.4**. Importantly, solutions from previous computations serve as initial guesses for subsequent computations, which decreases the computational costs significantly. While the aforementioned works focus on second order optimization methods, in the following, we apply the same philosophy to the first order method ISTA and develop a pathwise ISTA.

Algorithm 4 details the functionality of the pathwise ISTA. The pathwise ISTA is a *bottom-up* approach and starts off with $\mathbf{w}^{(0)} = \mathbf{0}$. As shown in **Appendix A**, $\mathbf{w}^{(0)} = \mathbf{0}$ is a stationary point of the underlying minimization problem for $\alpha^{(0)} = \max_i \left| \frac{\partial f}{\partial w_i}(\mathbf{0}) \right|$. In each step $l = 1, \dots, n_\alpha - 1$ of the pathwise ISTA, we successively decrease the regularization parameter according to $\alpha^{(l)} = (1 - \frac{l}{n_\alpha}) \alpha^{(0)}$ and solve the minimization problem using ISTA. At each step l , ISTA computes a sequence of solutions $\mathbf{w}^{(l)(k)}$ using the initial guess $\mathbf{w}^{(l)(k)} = \mathbf{w}^{(l-1)}$ until a convergence criterion is met. The converged solution $\mathbf{w}^{(l)}$ constitutes the solution corresponding to the regularization parameter $\alpha^{(l)}$ and serves as the initial guess for the subsequent step.

4. Automated material model discovery

The mathematical problems presented in **Section 2** constitute the backbone of the broad field of library-based material model discovery [22,25,26,28–30,56]. In the following, we draw a link to the problems in **Section 2** and automated material model discovery by introducing several example problems. In particular, we detail how the model-data-mismatch $f(\mathbf{w})$ can be formulated in the context

Algorithm 4 Pathwise iterative soft-thresholding algorithm (Pathwise ISTA).

Given f and n_α
 Choose the maximum number of steps NSTEP and convergence tolerance TOL
 $\alpha^{(0)} = \max_i \left| \frac{\partial f}{\partial w_i}(\mathbf{0}) \right|$
 $\alpha^{(l)} = (1 - \frac{1}{n_\alpha})\alpha^{(0)}$
 $\mathbf{w}^{(0)} = \mathbf{0}$
for $l = 1, \dots, n_\alpha - 1$ **do**
 $\alpha = \alpha^{(l)}$
 $\mathbf{w}^{(l)(0)} = \mathbf{w}^{(l-1)}$
 for $k = 0, \dots, \text{NSTEP} - 1$ **do**
 $\mathbf{w}^{(l)(k+1)} = \text{prox}_{\gamma g}(\mathbf{w}^{(l)(k)} - \gamma \nabla f(\mathbf{w}^{(l)(k)}))$
 if $\|\mathbf{w}^{(l)(k+1)} - \mathbf{w}^{(l)(k)}\|_2 < \text{TOL}$ **or** $|f(\mathbf{w}^{(l)(k+1)}) + g(\mathbf{w}^{(l)(k+1)}) - f(\mathbf{w}^{(l)(k)}) - g(\mathbf{w}^{(l)(k)})| < \text{TOL}$ **then**
 $\mathbf{w}^{(l)} = \mathbf{w}^{(l)(k+1)}$
 break
 end if
 end for
end for

of material modeling. We focus our attention on incompressible hyperelastic material models that either depend linearly or nonlinearly on the material parameters, see [86–88] for recent reviews, and consider labeled data from experiments with homogeneous deformation fields such as uniaxial tension or simple shear.

4.1. Material model library

In this work, we focus on library-based approaches for material model discovery, which constitute at the time the most prominent methods for material model discovery, see for example the works by [22,25,26,28–30,56]. That is, we formulate a general parametric ansatz for the material model and use the previously described methods to identify which of the parameters in the ansatz are necessary to describe the given data and which of the parameters may be set to zero. By identifying the most important parameters and setting others to zero, we arrive at a concise and thus interpretable mathematical expression of the material model.

A material model library for incompressible hyperelastic materials can be formulated by introducing a general parametric ansatz for the material's strain energy density function. Under the assumption of incompressibility and isotropy, the strain energy density W of a hyperelastic material is postulated as

$$W = \bar{W}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3; \mathbf{w}) - p \cdot [J - 1], \quad (39)$$

where $I_1 = \text{tr}(\mathbf{C})$, $I_2 = \frac{1}{2}(\text{tr}^2(\mathbf{C}) - \text{tr}(\mathbf{C}^2))$ are invariants of the right Cauchy-Green stretch tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$, $\mathbf{F}$ is the deformation gradient, $\lambda_1, \lambda_2, \lambda_3$ are the principal stretches defined as the eigenvalues of the right stretch tensor $\mathbf{U}$ obtained through the polar decomposition $\mathbf{F} = \mathbf{R}\mathbf{U}$, p is a scalar Lagrange multiplier that can be physically interpreted as the pressure, and $J = \det \mathbf{F} \stackrel{!}{=} 1$ is the determinant of the deformation gradient [89]. Our objective is to discover the function $\bar{W}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3; \mathbf{w})$ and its unknown parameters $\mathbf{w}$. Note that we explicitly indicate the dependence of the energy on both the invariants and the principal stretches, even though the invariants can be expressed as functions of the principal stretches, because we seek to design algorithms to automatically discover whether invariant-based or principal-stress-based models are superior to describe a given dataset.

The general ansatz $\bar{W}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3; \mathbf{w})$ comprises many well-known phenomenological material models, such as, for example, Mooney-Rivlin-type models or Ogden-type models. Existing material models can be broadly classified into models that depend linearly or nonlinearly on the material parameters $\mathbf{w}$. The classification of these models determines the structure of the optimization problem for determining the unknown parameters, see Problems 2.1,2.2 for the linear case and Problem 2.3 for the nonlinear case. Therefore, in the following, we treat these two cases separately and introduce two different material model libraries.

4.1.1. Linear material model library

Assuming that the model depends linearly on the parameters, the strain energy density may be written as

$$\bar{W}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3; \mathbf{w}) = \mathbf{Q}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3)^T \mathbf{w}, \quad (40)$$

where $\mathbf{Q} \in \mathbb{R}^m$ is a vector containing feature functions that depend on the invariants and principal stretches. We emphasize that, although $\bar{W}$ depends linearly on $\mathbf{w}$, the feature functions are nonlinear in general, which means that the model is still able to describe the nonlinear material responses of hyperelastic materials.

In this work, we consider the invariant-based generalized Mooney-Rivlin model [90–92] as the material model library

$$\bar{W}(I_1, I_2; \mathbf{w}) = \sum_{i=1}^{n_{\text{Mooney}}} \sum_{j=0}^i C_{ij} (I_1 - 3)^{i-j} (I_2 - 3)^j, \quad (41)$$

where n_{Mooney} defines the maximum polynomial order, and the material parameters C_{ij} can be collected into a vector $\mathbf{w}$ such that $\tilde{W}(I_1, I_2; \mathbf{w}) = \mathbf{Q}(I_1, I_2)^T \mathbf{w}$ with $\mathbf{Q}$ being a vector containing the nonlinear feature functions $(I_1 - 3)^{i-j}(I_2 - 3)^j$. The identification of the parameters in this model, without considering the L_1 -regularization, has been comprehensively studied by [93].

4.1.2. Nonlinear material model library

Some of the well-known material models used in practice depend nonlinearly on the material parameters, such as for example Ogden-type material models [94]. Thus, we consider the general material model library

$$\tilde{W}(I_1, I_2, \lambda_1, \lambda_2, \lambda_3; \mathbf{w}) = \sum_{i=1}^{n_{\text{Mooney}}} \sum_{j=0}^i C_{ij} [I_1 - 3]^{i-j} [I_2 - 3]^j + D [\lambda_1^\delta + \lambda_2^\delta + \lambda_3^\delta - 3], \quad (42)$$

which includes the previously described modeling features and an additional Ogden-type feature, which depends nonlinearly on the material parameters, see also [54]. The material parameter vector now comprises the parameters C_{ij} , D and δ .

4.2. Stress-strain relationship

The relationship between deformation and stress is obtained by differentiating the strain energy density. Specifically, the Piola stress $\mathbf{P}$ is computed by differentiating $\tilde{W}$ with respect to the deformation gradient

$$\mathbf{P} = \frac{\partial \tilde{W}}{\partial \mathbf{F}} = \frac{\partial \tilde{W}}{\partial \mathbf{F}} - p \mathbf{F}^{-T}, \quad (43)$$

where we used $\frac{\partial J}{\partial \mathbf{F}} = J \mathbf{F}^{-T}$. Applying the chain rule, we obtain

$$\frac{\partial \tilde{W}}{\partial F_{ij}} = \frac{\partial \tilde{W}}{\partial I_a} \frac{\partial I_a}{\partial F_{ij}} + \frac{\partial \tilde{W}}{\partial \lambda_b} \frac{\partial \lambda_b}{\partial F_{ij}}. \quad (44)$$

The derivatives of the strain energy density with respect to the strain invariants and the principal stretches can be computed analytically, or by means of automatic differentiation. Assuming $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \lambda_1$, the derivatives of the strain invariants and the principal stretches with respect to the deformation gradient are

$$\frac{\partial I_1}{\partial \mathbf{F}} = 2\mathbf{F}, \quad \frac{\partial I_2}{\partial \mathbf{F}} = 2I_1 \mathbf{F} - 2\mathbf{F}\mathbf{C}, \quad \frac{\partial \lambda_1}{\partial \mathbf{F}} = \mathbf{n}_1 \otimes \mathbf{N}_1, \quad \frac{\partial \lambda_2}{\partial \mathbf{F}} = \mathbf{n}_2 \otimes \mathbf{N}_2, \quad \frac{\partial \lambda_3}{\partial \mathbf{F}} = \mathbf{n}_3 \otimes \mathbf{N}_3, \quad (45)$$

where $\mathbf{N}_i$ denote the eigenvectors of $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ and $\mathbf{n}_i$ the eigenvectors of $\mathbf{b} = \mathbf{F} \mathbf{F}^T$ [89].

We focus on experiments with simple deformation fields for measuring stress-strain data pairs. Specifically, we put our attention on uniaxial compression/tension and simple shear. As we discuss in the following, under these load cases, the deformation gradient follows a specific structure, which simplifies the stress-strain relationship.

4.2.1. Uniaxial compression and tension

Experimental measurements of specimens under uniaxial compression and tension deliver labeled data pairs in the form (F_{11}, P_{11}) , where F_{11} and P_{11} are the longitudinal normal components of the deformation gradient and the Piola stress, respectively. As a result of the incompressibility assumption, $\det \mathbf{F} \stackrel{!}{=} 1$, and due to the symmetry condition, $F_{22} = F_{33}$, the deformation gradient under uniaxial compression/tension reads

$$\mathbf{F} = \begin{bmatrix} F_{11} & 0 & 0 \\ 0 & \frac{1}{\sqrt{F_{11}}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{F_{11}}} \end{bmatrix}. \quad (46)$$

To obtain a relationship between the longitudinal normal components of the deformation gradient and the Piola stress $P_{11}(F_{11}; \mathbf{w})$, the unknown hydrostatic pressure p in Eq. (43) needs to be computed. Using the zero-normal-stress condition, $P_{22} = P_{33} \stackrel{!}{=} 0$ along with Eq. (43), we find the hydrostatic pressure

$$P_{33} = \frac{\partial \tilde{W}}{\partial F_{33}} - p F_{33}^{-1} \stackrel{!}{=} 0, \quad \Rightarrow \quad p = \frac{\partial \tilde{W}}{\partial F_{33}} F_{33}. \quad (47)$$

We hence obtain the desired relationship by substituting the pressure in Eq. (43)

$$P_{11}(F_{11}; \mathbf{w}) = \frac{\partial \tilde{W}}{\partial F_{11}} - p F_{11}^{-1} = \frac{\partial \tilde{W}}{\partial F_{11}} - \frac{F_{33}}{F_{11}} \frac{\partial \tilde{W}}{\partial F_{33}}. \quad (48)$$

4.2.2. Simple shear

Experimental measurements of specimens under simple shear deliver labeled data pairs in the form (F_{12}, P_{12}) , where F_{12} and P_{12} are the shear components of the deformation gradient and the Piola stress, respectively. The deformation gradient under simple shear simplifies to

$$\mathbf{F} = \begin{bmatrix} 1 & F_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{F}^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ -F_{12} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (49)$$

which satisfies the incompressibility constraint. Because the shear component of the transposed inverse of the deformation gradient vanishes $\{\mathbf{F}_{SS}^{-T}\}_{12} = 0$, the relationship $P_{12}(F_{12}; \mathbf{w})$ does not depend on the pressure p , and we obtain

$$P_{12}(F_{12}; \mathbf{w}) = \frac{\partial \tilde{W}}{\partial F_{12}}. \quad (50)$$

4.3. Model-data-mismatch

To infer information about the material parameters $\mathbf{w}$, we base in this work on uniaxial tension/compression data in the form of data pairs $(F_{11}^{(i)}, P_{11}^{(i)})$ with $i = 1, \dots, n_{\text{UTC}}$, while the simple shear data takes the form $(F_{12}^{(j)}, P_{12}^{(j)})$ with $j = 1, \dots, n_{\text{SS}}$. We define $P_{11}^{\max} = \max_i |P_{11}^{(i)}|$ and $P_{12}^{\max} = \max_j |P_{12}^{(j)}|$ and choose the model-data-mismatch for the nonlinear material model library in Section 4.1.2 as

$$f(\mathbf{w}) = \frac{1}{2[n_{\text{UTC}} + n_{\text{SS}}]} \left[\sum_{i=1}^{n_{\text{UTC}}} \left[\frac{P_{11}(F_{11}^{(i)}; \mathbf{w}) - P_{11}^{(i)}}{P_{11}^{\max}} \right]^2 + \sum_{j=1}^{n_{\text{SS}}} \left[\frac{P_{12}(F_{12}^{(j)}; \mathbf{w}) - P_{12}^{(j)}}{P_{12}^{\max}} \right]^2 \right], \quad (51)$$

which provides a metric for quantifying the mismatch between the model predictions and the data. Due to the division by $P_{ij}^{\max}$, the model-data-mismatch is non-dimensionalized. In this way, the contributions from the uniaxial tension/compression data and the simple shear data exert an equal influence on the model-data mismatch.

The model-data-mismatch for the linear material model library in Section 4.1.1 is defined similarly. However, we additionally consider a normalization of the feature vectors. To this end, we assemble the feature matrix

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{X}}^{\text{UTC}} \\ \tilde{\mathbf{X}}^{\text{SS}} \end{bmatrix}, \quad \text{with} \quad \tilde{X}_{ij}^{\text{UTC}} = \frac{1}{P_{11}^{\max}} \left[\frac{\partial Q_j}{\partial F_{11}}(F_{11}^{(i)}) - \frac{F_{33}^{(i)}}{F_{11}^{(i)}} \frac{\partial Q_j}{\partial F_{33}}(F_{11}^{(i)}) \right], \quad \tilde{X}_{ij}^{\text{SS}} = \frac{1}{P_{12}^{\max}} \frac{\partial Q_j}{\partial F_{12}}(F_{12}^{(i)}), \quad (52)$$

where we considered Eqs. (48) and (50), and the measurement vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{\text{UTC}} \\ \mathbf{y}^{\text{SS}} \end{bmatrix}, \quad \text{with} \quad y_i^{\text{UTC}} = \frac{P_{11}^{(i)}}{P_{11}^{\max}}, \quad y_i^{\text{SS}} = \frac{P_{12}^{(i)}}{P_{12}^{\max}}. \quad (53)$$

Then, we assemble a normalized feature matrix $\mathbf{X}$, whose columns are computed as $\mathbf{X}_i = \tilde{\mathbf{X}}_i / \|\tilde{\mathbf{X}}_i\|_2$, see Section 2.1. The model-data-mismatch for the linear material model library is finally defined as

$$f(\mathbf{w}) = \frac{1}{2[n_{\text{UTC}} + n_{\text{SS}}]} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2, \quad (54)$$

where we note that, after using this model-data-mismatch for CD or LARS-LASSO, the material parameters must be rescaled according to $\tilde{\mathbf{w}}_i = \mathbf{w}_i / \|\tilde{\mathbf{X}}_i\|_2$, in which $\tilde{\mathbf{w}}_i$ are the actual material model parameters.

5. Benchmarking

In the following, we apply the discussed algorithms to different benchmark problems. We will distinguish between the material models that depend linearly and nonlinearly on the material parameters.

5.1. Linear material model library

First, we consider material models that depend linearly on the material parameters. We consider four benchmark material models; the Neo-Hookean model, the Mooney-Rivlin model, the Yeoh model, and the Biderman model. The models' strain energy density functions and the values of the material parameters are shown in Table 3. We generate synthetic data for the four material models by choosing n_{UTC} equidistant values between 0.75 and 1.5 for $F_{11}^{(i)}$ in the uniaxial tension/compression case, and n_{SS} equidistant values between 0 and 0.5 for $F_{12}^{(j)}$ in the simple shear case. We consider both the noise-free data as well as data perturbed by independent Gaussian noise, i.e., $P_{ij}^{(i)} = P_{ij}^{(i)} + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma)$. Specifically, we choose a standard deviation of $\sigma = 5$, resulting in an exceptionally high noise in the data.

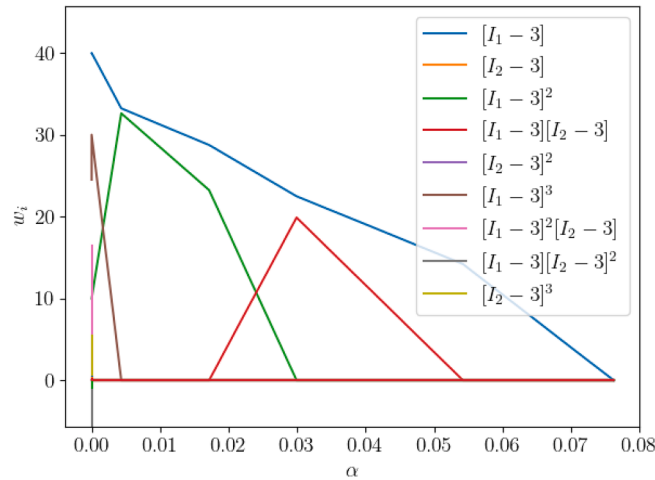
After generating all benchmark datasets, we apply CD for solving Problem 2.1 assuming the linear material model library defined in Section 4.1.1 considering Mooney-Rivlin features up to a polynomial order of four. Due to the L_1 -norm regularization, the CD algorithm results in sparse material parameter vectors and thus concise mathematical expressions for the strain energy density. The L_1 -norm regularization not only drives certain parameters to zero, but also induces shrinkage in the remaining nonzero parameters. Therefore, in a postprocessing step, we use the features identified as active by the CD algorithm to solve an unregularized regression problem, i.e., Problem 2.1 considering only the active features and $\alpha = 0$. This further decreases the model-data-mismatch while leaving the material model unchanged.

Table 3 shows the material models discovered through the CD algorithm after the postprocessing step. For all benchmarks except for the Biderman model, the correct model is discovered in the noise-free case. For the Neo-Hookean and Mooney-Rivlin models, the correct model is also discovered for the noisy data. For all other cases, surrogate models with satisfactory fitting accuracy and sparsity are discovered.

Table 3

Strain energy density functions of the true and discovered material models and model-data-mismatch.

Benchmarks		Strain energy density $\bar{W}$	$f(\mathbf{w})$
Neo-Hookean	Truth	$40.00[I_1 - 3]$	–
	$\sigma = 0$	$40.00[I_1 - 3]$	$7.34 \cdot 10^{-33}$
	$\sigma = 5$	$40.22[I_1 - 3]$	0.0040
Mooney-Rivlin	Truth	$40.00[I_1 - 3] + 20.00[I_2 - 3]$	–
	$\sigma = 0$	$40.00[I_1 - 3] + 20.00[I_2 - 3]$	$3.76 \cdot 10^{-32}$
	$\sigma = 5$	$46.90[I_1 - 3] + 12.94[I_2 - 3]$	0.0018
Yeoh	Truth	$40.00[I_1 - 3] + 10.00[I_1 - 3]^2 + 30.00[I_1 - 3]^3$	–
	$\sigma = 0$	$40.00[I_1 - 3] + 10.00[I_1 - 3]^2 + 30.00[I_1 - 3]^3$	$3.84 \cdot 10^{-32}$
	$\sigma = 5$	$33.76[I_1 - 3] + 40.77[I_1 - 3]^2$	0.0020
Biderman	Truth	$40.00[I_1 - 3] + 20.00[I_2 - 3] + 10.00[I_1 - 3]^2 + 30.00[I_1 - 3]^3$	–
	$\sigma = 0$	$55.62[I_1 - 3] + 20.21[I_1 - 3]^2 + 12.92[I_1 - 3][I_2 - 3]$	$1.98 \cdot 10^{-4}$
	$\sigma = 5$	$53.14[I_1 - 3] + 14.41[I_1 - 3]^2 + 27.80[I_2 - 3]^2$	0.0011

**Fig. 3.** Regularization path computed by LARS-LASSO for the noise-free Yeoh dataset. For clarity, legend entries of higher order features are omitted.

An in-depth discussion is required for the noise-free Biderman benchmark. Two potential factors may prevent the exact model from being recovered. First, for the considered experimental setup, there may exist multiple material models in the model library that exhibit identical or nearly identical stress responses. This is evidenced by the fact that the discovered material models – despite differing from the ground truth models – show excellent agreement with the data. And second, solving the L_1 -regularized problem does not guarantee finding the best material model in the model library. This is due to the approximation of the L_0 -pseudo norm by the L_1 -regularization term.

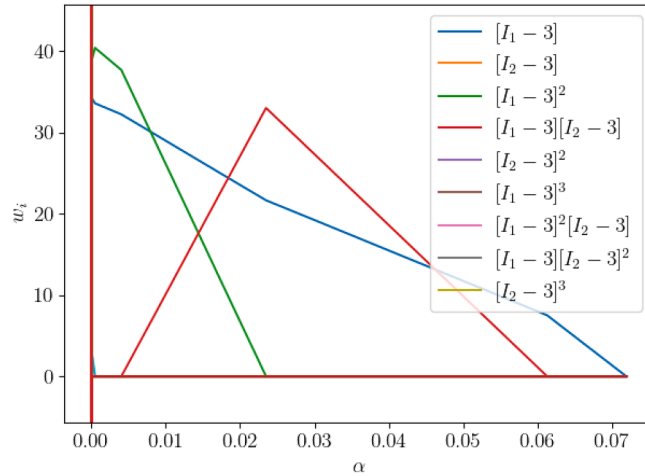
We note that ISTA can readily be applied to [Problem 2.1](#). However, when $f(\mathbf{w})$ is quadratic, the CD algorithm provides a more efficient solution strategy, as it directly exploits the closed-form solution of the one-dimensional subproblems along each coordinate. Moreover, CD does not require the selection of a step-size parameter. Our numerical tests show that, in general, CD reaches the same convergence criterion approximately two orders of magnitude faster than ISTA when applied to the same optimization problem. We therefore find that CD should be preferred over ISTA in cases where $f(\mathbf{w})$ is quadratic.

One deficiency of using CD for solving [Problem 2.1](#), is that it is not known a priori how the regularization parameter α must be chosen to obtain a sparse material model with low model-data-mismatch. A suitable choice for the regularization parameter can for example be found through a manual trial-and-error procedure. On the contrary, previous works [\[95\]](#) have proposed automated strategies for choosing the value α in [Problem 2.1](#). These selection strategies require to repeatedly solve the problem for different values of α . In this work, we investigate a third option, and apply LARS-LASSO for computing the regularization path of [Problem 2.1](#). We will demonstrate that knowing the regularization path simplifies the selection of a suitable value of α .

Table 4

First steps of LARS-LASSO for the noise-free Yeoh dataset.

Step	Strain energy density $\tilde{W}$	α	$f(\mathbf{w})$
0	0.00	$7.62 \cdot 10^{-2}$	$1.21 \cdot 10^{-1}$
1	$14.21[I_1 - 3]$	$5.42 \cdot 10^{-2}$	$6.31 \cdot 10^{-2}$
2	$22.49[I_1 - 3] + 19.88[I_1 - 3][I_2 - 3]$	$2.99 \cdot 10^{-2}$	$1.99 \cdot 10^{-2}$
3	$28.74[I_1 - 3] + 23.24[I_1 - 3]^2$	$1.72 \cdot 10^{-2}$	$6.60 \cdot 10^{-3}$
4	$33.24[I_1 - 3] + 32.63[I_1 - 3]^2$	$2.32 \cdot 10^{-3}$	$6.14 \cdot 10^{-4}$
5	$40.00[I_1 - 3] + 10.00[I_1 - 3]^2 + 30.00[I_1 - 3]^3$	$5.43 \cdot 10^{-16}$	$9.21 \cdot 10^{-30}$

**Fig. 4.** Regularization path computed by LARS-LASSO for the noisy Yeoh dataset. For clarity, legend entries of higher order features are omitted.

To demonstrate the functionality of LARS-LASSO, we show the computed regularization path in Fig. 3 as well as the first steps of LARS-LASSO in Table 4, both for the noise-free Yeoh dataset. It is observed that after five iterations, the model-data-mismatch decreases to effectively zero, as LARS-LASSO has discovered the ground truth material model by identifying the correct modeling features.

Notably, LARS-LASSO does not only consider correct features during the first iterations. After the second iteration, LARS-LASSO identifies a false-positive feature, i.e., a feature that appears in the discovered model while not appearing in the Yeoh model. However, after the third iteration, this feature is eliminated. This behavior can be traced back to the modification applied to LARS in Section 3.2.3.

At this point, it is important to mention that LARS-LASSO computes the knots of the regularization path. Consequently, not each step of LARS-LASSO corresponds to a critical value α_c as defined in Problem 2.2, see Fig. 1b. However, for a given set of knots, the critical values can be easily extracted. For example, in the second to fourth steps in Table 4, the parameter vectors exhibit the same number of nonzero parameters. Therefore, out of these three steps, only the fourth step corresponds to a critical value. Furthermore, as only the initial steps of LARS-LASSO are of interest in practice, and due to the early stopping criterion described in Appendix C.2.2, LARS-LASSO does not identify all critical values, but only the practically relevant initial ones.

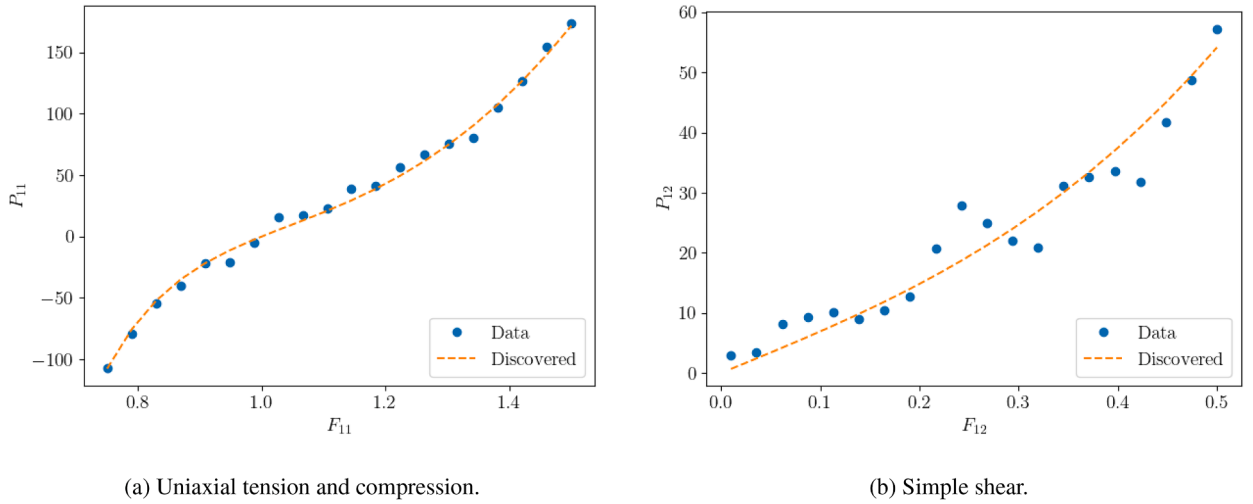
Table 4 highlights the important advantages of LARS-LASSO over methods like CD for solving Problem 2.1. When approaching Problem 2.1 with the CD algorithm, a suitable value for α must be found either through trial-and-error or by solving the problem for multiple values of α and conducting a Pareto analysis [95]. LARS-LASSO always starts with the smallest value of α for which all parameters are zero and then subsequently decreases α . Importantly, α is not decreased in equidistant steps, but with varying step sizes. The algorithm guarantees that no significant changes occur between two consecutive steps, i.e., between two steps no features are added or removed from the active set. In this way, LARS-LASSO efficiently identifies the critical and practically meaningful values for α . For example, between the fourth and fifth steps, α decreases by several orders of magnitude, as LARS-LASSO automatically identifies that no significant changes occur across these orders of magnitude. Choosing values of α that are between two LARS-LASSO steps is practically not meaningful, as there exists a smaller value of α yielding the same material model. LARS-LASSO thus identifies intervals of α that have no impact on the model and can be disregarded.

Next, we consider the noisy dataset corresponding to the Yeoh model. Fig. 4 and Table 5 show the regularization path identified by LARS-LASSO. After the fourth step of LARS-LASSO, the model-data-mismatch is barely decreasing. Therefore, $\alpha = 4.91 \cdot 10^{-4}$ is a good choice for this example. Notably, as a result of the noise, LARS-LASSO does not discover the Yeoh model, but a surrogate model that fits the dataset while being expressed as a short mathematical expression (see Fig. 5).

Table 5

First steps of LARS-LASSO for the noisy Yeoh dataset.

Step	Strain energy density $\bar{W}$	α	$f(\mathbf{w})$
0	0.00	$7.19 \cdot 10^{-2}$	0.1102
1	$7.52[I_1 - 3]$	$6.13 \cdot 10^{-2}$	0.0819
2	$21.65[I_1 - 3] + 33.00[I_1 - 3][I_2 - 3]$	$2.35 \cdot 10^{-2}$	0.0142
3	$32.21[I_1 - 3] + 37.66[I_1 - 3]^2$	$4.03 \cdot 10^{-3}$	0.0023
4	$33.57[I_1 - 3] + 40.39[I_1 - 3]^2$	$4.91 \cdot 10^{-4}$	0.0020
5	$34.15[I_1 - 3] + 39.07[I_1 - 3]^2 + 2.58[I_1 - 3]^4$	$1.16 \cdot 10^{-4}$	0.0020

**Fig. 5.** Stress-strain response of the discovered material model for the noisy Yeoh dataset.

We apply LARS-LASSO in the same manner to all datasets. For each benchmark, we choose a LARS-LASSO step at which the solution is sparse and the model-data-mismatch is low, and afterwards apply the previously described postprocessing step to further reduce the model-data-mismatch. The discovered material models are equal to those discovered with the CD algorithm, see Table 3.

5.2. Nonlinear material model library

Next, we consider benchmark problems with material models that depend nonlinearly on the parameters, and apply ISTA for material model discovery. Specifically, we choose the Mooney-Rivlin model, the Ogden model and a mixed model that contains both the Mooney-Rivlin and Ogden features, see Table 6. We generate synthetic data in a way analogous to Section 5.1 and perturb them by independent Gaussian noise.

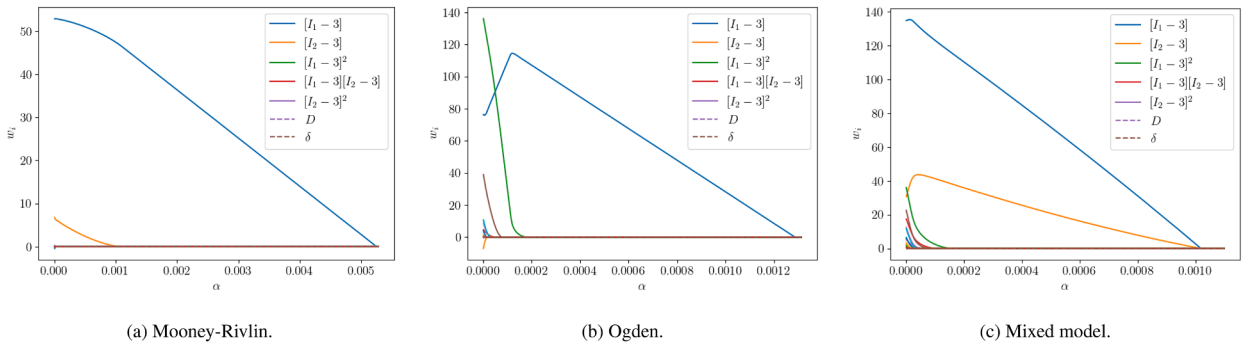
We apply ISTA assuming the nonlinear material model library defined in Section 4.1.2. For all benchmarks, we choose $\mathbf{w}^{(0)} = \mathbf{1}$ as the initial guess to demonstrate the sparsity-promoting property of the L_1 -regularization term. As described earlier, after ISTA has converged and a material model with several vanishing material parameters has been discovered, as a postprocessing step, we solve the unregularized Problem 2.3 while keeping the zero parameters fixed. Again, this is motivated in further reducing the model-data-mismatch while the model remains unchanged. Table 6 shows the material models discovered for different noise levels. For the Ogden model, ISTA discovers the correct material models both in the noise-free and the noisy cases, and the discovered models exhibit a small model-data-mismatch. For the Mooney-Rivlin model and the mixed model, however, despite showing a low model-data-mismatch, the discovered material models do not match the ground truth models. As discussed previously, this can be a result of the approximation of the L_0 -pseudo norm by the L_1 -regularization term, which does not guarantee that the best model in the library is found. Additionally, there may be similar features in the model library, which makes it difficult to recover the exact material model in the inverse problem. In fact, the first Mooney-Rivlin feature is equal to the Ogden feature for $\delta = 2$, i.e., $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$. This could explain why ISTA discovers the Ogden feature instead of the first Mooney-Rivlin feature. In practical applications, the ground truth model is generally unavailable. Therefore, the primary objective is typically to identify a sparse model that exhibits minimal discrepancy between the model and the observed data, an objective that ISTA is well-suited to achieve.

Finally, we use the pathwise ISTA with $n_\alpha = 1000$ to approximately compute the regularization path. Fig. 6 shows the results for the noisy datasets. As α decreases, the number of nonzero material parameters and the fitting accuracy of the models increase. The pathwise ISTA correctly identifies the Mooney-Rivlin features for the Mooney-Rivlin model and the mixed model. However, for the reasons discussed above, it fails to identify the Ogden feature in both the Ogden and mixed models, instead selecting surrogate

Table 6

Strain energy density functions of the true and discovered material models and model-data-mismatch.

Benchmarks		Strain energy density $\bar{W}$	$f(\mathbf{w})$
Mooney-Rivlin	Truth	$40.00[I_1 - 3] + 20.00[I_2 - 3]$	–
	$\sigma = 0$	$21.07[I_2 - 3] + 24.61[\lambda_1^{2.48} + \lambda_2^{2.48} + \lambda_3^{2.48} - 3]$	$4.28 \cdot 10^{-05}$
	$\sigma = 5$	$17.22[I_2 - 3] + 23.74[\lambda_1^{2.64} + \lambda_2^{2.64} + \lambda_3^{2.64} - 3]$	0.0019
Ogden	Truth	$5.00[\lambda_1^{8.00} + \lambda_2^{8.00} + \lambda_3^{8.00} - 3]$	–
	$\sigma = 0$	$4.94[\lambda_1^{8.03} + \lambda_2^{8.03} + \lambda_3^{8.03} - 3]$	$6.21 \cdot 10^{-07}$
	$\sigma = 5$	$4.99[\lambda_1^{8.04} + \lambda_2^{8.04} + \lambda_3^{8.04} - 3]$	0.0003
Mixed Model	Truth	$40.00[I_1 - 3] + 20.00[I_2 - 3] + 5.00[\lambda_1^{8.00} + \lambda_2^{8.00} + \lambda_3^{8.00} - 3]$	–
	$\sigma = 0$	$12.85[\lambda_1^{6.54} + \lambda_2^{6.54} + \lambda_3^{6.54} - 3]$	$5.57 \cdot 10^{-05}$
	$\sigma = 5$	$13.19[\lambda_1^{6.51} + \lambda_2^{6.51} + \lambda_3^{6.51} - 3]$	0.0005

**Fig. 6.** Regularization path computed by the pathwise ISTA for the noisy datasets. For clarity, legend entries of higher order Mooney-Rivlin features are omitted.

features that mimic the behavior of the true Ogden feature. We note that, due to the different initial guesses, the regularization path may not contain the models discovered by ISTA shown in Table 6. Specifically, for the results in Table 6, we assumed the initial guess $\mathbf{w}^{(0)} = \mathbf{1}$, while the initial guess in each step of the pathwise ISTA depends on the result of the previous step.

6. Conclusions and outlook

The field of non-smooth optimization provides a range of tools for addressing challenges in material model discovery. In this work, we have discovered material models that depend linearly or nonlinearly on the material parameters using the CD algorithm and ISTA, respectively. These methods robustly solve the underlying sparse regression problems with proven convergence for a given value of the regularization parameter. Using the CD algorithm is preferred over ISTA when the material model depends linearly on the material parameters, as CD does not require step size selection and efficiently leverages the closed-form solution of the corresponding one-parameter problem. For material models that depend nonlinearly on the material parameters, however, ISTA offers an attractive alternative with proven convergence if the step size is sufficiently small. Both the CD algorithm and ISTA require choosing the regularization parameter a priori. Conversely, the LARS-LASSO algorithm leverages the piecewise linearity of the regularization path to efficiently compute the critical values of the regularization parameter at which the number of nonzero elements in the material parameter vector changes. This facilitates the manual selection of a material model from the first steps of LARS-LASSO. For mechanics applications, where we are typically interested in material models with only a few nonzero parameters, LARS-LASSO offers an efficient alternative to the CD algorithm – especially when it is terminated after the first few iterations to avoid computing the full regularization path. Finally, by successively decreasing the regularization parameter and using the previous solutions as initial guesses for the subsequent solves, the novel pathwise ISTA efficiently computes the nonlinear regularization path when the material models depend nonlinearly on the material parameters. The concepts presented in this work can be extended in several directions. For example, we did not consider constraints on the material parameters. Such constraints can be incorporated into the CD algorithm, LARS-LASSO, and ISTA without significant effort. Additionally, for material models that depend nonlinearly on the material parameters, we have focused on the first-order method ISTA. In the future, second-order methods that exploit the

second derivative of the model-data mismatch may be explored for material model discovery. In the future, we aim to investigate all algorithms discussed in this paper in the context of dissipative materials and to apply the discussed algorithms to experimentally measured data.

CRedit authorship contribution statement

Moritz Flaschel: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Trevor Hastie:** Writing – review & editing, Supervision, Conceptualization; **Ellen Kuhl:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Code and data availability

Code and data are publicly available on Zenodo (<https://doi.org/10.5281/zenodo.15848305>), see [96], and on GitHub at <https://github.com/mflaschel/non-smooth-material-model-discovery>.

Data availability

All codes and data are publically available.

Declaration of competing interest

The authors, Moritz Flaschel, Trevor Hastie, and Ellen Kuhl, declare no conflict of interest regarding the manuscript entitled “Non-smooth optimization meets automated material model discovery”.

Acknowledgment

Moritz Flaschel thanks Clemens Sirotenko for fruitful discussions and literature suggestions. This work was supported by the ERC Advanced Grant 101141626 DISCOVER to Ellen Kuhl. The authors utilized HAWKI, a large language model interface provided by FAU, to enhance the writing style in certain sections of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix A. Boundedness of the regularization parameter

Choosing a large value of α in [Problem 2.1](#) or [Problem 2.3](#) may result in a solution vector whose entries are all zero, i.e., $\mathbf{w}^* = \mathbf{0}$. As we are typically not interested in this zero solution, we seek to choose smaller values of α in practice. We are thus interested in determining the minimum value of α such that $\mathbf{w}^* = \mathbf{0}$, which we will denote by $\alpha^{(0)}$. To this end, we consider the necessary condition for a minimum of [Problem 2.3](#)

$$0 \in \frac{\partial f}{\partial w_i}(\mathbf{0}) + \alpha^{(0)}[-1, 1], \quad (\text{A.1})$$

which must hold true for all i . After shifting the partial derivative of f to the left and taking the absolute value, we obtain

$$\left| \frac{\partial f}{\partial w_i}(\mathbf{0}) \right| \in \alpha^{(0)}[0, 1]. \quad (\text{A.2})$$

The minimum value of α that fulfills this condition for all i is

$$\alpha^{(0)} = \max_i \left| \frac{\partial f}{\partial w_i}(\mathbf{0}) \right|. \quad (\text{A.3})$$

In practice, we are interested in values of α that do not yield the zero solution, i.e.,

$$\alpha \in (0, \alpha^{(0)}). \quad (\text{A.4})$$

We note that [Eq. \(A.1\)](#) is a necessary but not sufficient condition. Thus, dependent on the appearance of f , $\alpha^{(0)}$ may not necessarily yield the zero solution. Nevertheless, we will adhere to the above range of α in this work.

Finally, for the special case in [Problem 2.1](#), we obtain [\[49\]](#)

$$\alpha^{(0)} = \frac{1}{n} \max_i |\mathbf{X}_i^T \mathbf{y}|. \quad (\text{A.5})$$

Appendix B. Additional information to LARS and LARS-LASSO

B.1. Boundedness of the step size

It can be shown that the step size of LARS, see Eq. (B.1), is bounded by $0 < \gamma^{(k)} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$ [45]. To show it, we define

$$\gamma_{j^*}^{(k)} = \min^+ \left\{ \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}}, \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}} \right\}, \quad (\text{B.1})$$

and show that $0 < \gamma_{j^*}^{(k)} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$ for all $j^* \in \mathcal{A}^{(k)}$. First, we observe that $\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)} > 0$ and $\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)} > 0$, and thus $\gamma_{j^*}^{(k)} > 0$. Next, we distinguish the cases $a_{j^*}^{(k)} < 0$ and $a_{j^*}^{(k)} > 0$, where the case $a_{j^*}^{(k)} = 0$ is apparent.

- Case 1: We consider the case $a_{j^*}^{(k)} < 0$. We further distinguish the sub-cases $A^{(k)} + a_{j^*}^{(k)} < 0$ and $A^{(k)} + a_{j^*}^{(k)} > 0$.
 - Sub-case 1.1: We consider the sub-case $A^{(k)} + a_{j^*}^{(k)} < 0$. It follows $\frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}} < 0$. Thus, $\gamma_{j^*}^{(k)} = \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$.
 - Sub-case 1.2: We consider the sub-case $A^{(k)} + a_{j^*}^{(k)} > 0$. It follows $0 < \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}} < \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}}$. Thus, $\gamma_{j^*}^{(k)} = \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$.
- Case 2: We consider the case $a_{j^*}^{(k)} > 0$. We further distinguish the sub-cases $A^{(k)} - a_{j^*}^{(k)} < 0$ and $A^{(k)} - a_{j^*}^{(k)} > 0$.
 - Sub-case 2.1: We consider the sub-case $A^{(k)} - a_{j^*}^{(k)} < 0$. It follows $\frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}} < 0$. Thus, $\gamma_{j^*}^{(k)} = \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}}$. We notice that $\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)} < 2\bar{c}_{\max}^{(k)}$ and $A^{(k)} + a_{j^*}^{(k)} > 2A^{(k)}$. Therefore, $\gamma_{j^*}^{(k)} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$.
 - Sub-case 2.2: We consider the sub-case $A^{(k)} - a_{j^*}^{(k)} > 0$. We further distinguish the sub-sub-cases $a_{j^*}^{(k)} < \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} c_{j^*}^{(k)}$ and $a_{j^*}^{(k)} > \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} c_{j^*}^{(k)}$.
 For $a_{j^*}^{(k)} < \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} c_{j^*}^{(k)}$, it is $0 < \frac{\bar{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{A^{(k)} - a_{j^*}^{(k)}} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$, and for $a_{j^*}^{(k)} > \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} c_{j^*}^{(k)}$, it is $0 < \frac{\bar{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{A^{(k)} + a_{j^*}^{(k)}} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$. Therefore, $\gamma_{j^*}^{(k)} < \frac{\bar{c}_{\max}^{(k)}}{A^{(k)}}$.

B.2. How inactive parameters enter the active set

Inactive parameters can be shown to enter the active set such that their sign is equal to the sign of the correlation of the corresponding feature vector [45]. Formally, this can be formulated as follows. We consider a step k in which one parameter w_l enters the active set, i.e., $w_l^{(k)} = 0$ and $w_l^{(k+1)} \neq 0$, with l such that $l \notin \mathcal{A}^{(k-1)}$ and $l \in \mathcal{A}^{(k)} = \mathcal{A}^{(k-1)} \cup \{l\}$. It can be shown that $\text{sign}(w_l^{(k+1)}) = \text{sign}(c_l^{(k)})$.

We first consider the first step $k = 0$ for which the active set reduces to $\mathcal{A}^{(0)} = \{l\}$. It is $w_l^{(1)} = \gamma^{(0)} \bar{c}_u^{(0)} \text{sign}(c_l^{(0)})$, where we used $w_l^{(0)} = 0$ and $\bar{X}_l^{T(0)} \bar{X}_l^{(0)} = 1$. Because $\gamma^{(0)} > 0$, $\bar{c}_u^{(0)} > 0$, it is $\text{sign}(w_l^{(1)}) = \text{sign}(c_l^{(0)})$.

Next, we focus on the general case $k > 1$. We recall that $\mathbf{w}_{\mathcal{A}}^{(k+1)} = \mathbf{w}_{\mathcal{A}}^{(k)} + \Delta \mathbf{w}_{\mathcal{A}}^{(k)}$, where we have $w_l^{(k)} = 0$ for the element of interest, such that $w_l^{(k+1)} = \Delta w_l^{(k)}$. The parameter updates are computed according to

$$\Delta \mathbf{w}_{\mathcal{A}}^{(k)} = \gamma^{(k)} A^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \mathbf{1}_{\mathcal{A}}. \quad (\text{B.2})$$

Because all feature vectors of the active set share the same correlation in absolute value, we may substitute

$$\mathbf{1}_{\mathcal{A}} = \frac{\bar{c}_{\max}^{(k)}}{\bar{c}_{\max}^{(k)}} \mathbf{1}_{\mathcal{A}} = \frac{1}{\bar{c}_{\max}^{(k)}} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} [\mathbf{y}_{\parallel} - \boldsymbol{\mu}^{(k)}], \quad (\text{B.3})$$

to obtain

$$\Delta \mathbf{w}_{\mathcal{A}}^{(k)} = \gamma^{(k)} \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) \underbrace{\left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)}}_{\mathbf{w}_{\parallel}^{(k)}} [\mathbf{y}_{\parallel} - \boldsymbol{\mu}^{(k)}], \quad (\text{B.4})$$

where we identify $\mathbf{w}_{\parallel}^{(k)}$ as the least squares solution $\mathbf{w}_{\parallel}^{(k)} = \arg \min_{\mathbf{w}_{\parallel}} \|\bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \mathbf{w}_{\parallel} - \mathbf{r}_{\parallel}^{(k)}\|^2$. It is $\Delta w_l^{(k)} = \gamma^{(k)} \frac{A^{(k)}}{\bar{c}_{\max}^{(k)}} \text{sign}(c_l^{(k)}) w_{\parallel}^{(k)}$, and therefore $\text{sign}(w_l^{(k+1)}) = \text{sign}(c_l^{(k)})$ if $w_{\parallel}^{(k)} > 0$.

Thus, the remaining task is to show that $w_{\parallel}^{(k)}$ is positive. To this end, we additively divide $\mathbf{y}_{\parallel}$ into the two contributions $\mathbf{y}_{\parallel}^{(k-1)}$ and $\mathbf{y}_{\parallel} - \mathbf{y}_{\parallel}^{(k-1)}$. Specifically, we define $\mathbf{y}_{\parallel}^{(k-1)} = \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \mathbf{y}$, which we may write as

$$\begin{aligned} \mathbf{y}_{\parallel}^{(k-1)} &= \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} [\mathbf{y}_{\parallel} + \mathbf{y}_{\perp} + \boldsymbol{\mu}^{(k-1)} - \boldsymbol{\mu}^{(k-1)}] \\ &= \boldsymbol{\mu}^{(k-1)} + \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} [\mathbf{y}_{\parallel} - \boldsymbol{\mu}^{(k-1)}] \\ &= \boldsymbol{\mu}^{(k-1)} + \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \right]^{-1} \bar{c}_{\max}^{(k-1)} \mathbf{1}_{\mathcal{A}} \\ &= \boldsymbol{\mu}^{(k-1)} + \frac{\bar{c}_{\max}^{(k-1)}}{A^{(k-1)}} \mathbf{u}^{(k-1)}, \end{aligned} \quad (\text{B.5})$$

where we used $\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \mathbf{y}_{\perp} = \mathbf{0}$, $\boldsymbol{\mu}^{(k-1)} = \mathbf{X} \mathbf{w}^{(k-1)} = \bar{\mathbf{X}}_{\mathcal{A}}^{(k-1)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k-1)}) \mathbf{w}_{\mathcal{A}}^{(k-1)}$, and $\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} \mathbf{r}_{\parallel}^{(k-1)} = \bar{c}_{\max}^{(k-1)} \mathbf{1}_{\mathcal{A}}$.

Recalling that $\boldsymbol{\mu}^{(k)} = \boldsymbol{\mu}^{(k-1)} + \gamma^{(k-1)} \mathbf{u}^{(k-1)}$, we can rewrite $\mathbf{w}_{\parallel}^{(k)}$ as

$$\begin{aligned} \mathbf{w}_{\parallel}^{(k)} &= \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} [\mathbf{y}_{\parallel} - \mathbf{y}_{\parallel}^{(k-1)} + \mathbf{y}_{\parallel}^{(k-1)} - \boldsymbol{\mu}^{(k-1)} + \gamma^{(k-1)} \mathbf{u}^{(k-1)}] \\ &= \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \left[\mathbf{y}_{\parallel} - \mathbf{y}_{\parallel}^{(k-1)} + \left(\frac{\bar{c}_{\max}^{(k-1)}}{A^{(k)}} + \gamma^{(k-1)} \right) \mathbf{u}^{(k-1)} \right] \\ &= \underbrace{\left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} [\mathbf{y}_{\parallel} - \mathbf{y}_{\parallel}^{(k-1)}]}_{\mathbf{w}_{\mathbf{y}}^{(k)}} + \underbrace{\left(\frac{\bar{c}_{\max}^{(k-1)}}{A^{(k)}} + \gamma^{(k-1)} \right) \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \mathbf{u}^{(k-1)}}_{\mathbf{w}_{\mathbf{u}}^{(k)}}. \end{aligned} \quad (\text{B.6})$$

By the definition of $\mathbf{y}_{\parallel}^{(k-1)}$, it is $\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)} [\mathbf{y} - \mathbf{y}_{\parallel}^{(k-1)}] = \mathbf{0}$. Consequently, we observe that $\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} [\mathbf{y} - \mathbf{y}_{\parallel}^{(k-1)}]$ must be zero for all $i \in \mathcal{A}^{(k-1)}$ and can only be nonzero for $i = l$. We define the vector $\boldsymbol{\delta}$ such that $\delta_i = 0$ if $i \neq l$ and $\delta_l = \bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} [\mathbf{y} - \mathbf{y}_{\parallel}^{(k-1)}]$. Thus, it is $\mathbf{w}_{\mathbf{y}}^{(k)} = \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \boldsymbol{\delta}$, where the element of interest is $w_{\mathbf{y} \, l}^{(k)} = \left\{ \left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1} \right\}_{ll} \delta_l$, where no sum of repeated indices is applied. We find that $w_{\mathbf{y} \, l}^{(k)}$ is positive because the diagonal elements of the positive definite matrix $\left[\bar{\mathbf{X}}_{\mathcal{A}}^{T(k)} \bar{\mathbf{X}}_{\mathcal{A}}^{(k)} \right]^{-1}$ are positive and δ_l is positive, see [45]. Finally, $w_{\mathbf{u} \, l}^{(k)}$ vanishes because $\mathbf{u}^{(k-1)}$ is a linear combination of the feature vectors $\bar{\mathbf{X}}_{\mathcal{A}}^{T(k-1)}$.

B.3. Computation of the regularization parameter

The solutions $\mathbf{w}^{(k)}$ obtained by LARS-LASSO are solutions to Problem 2.1 [45] for different choices of the regularization parameter. This raises the question of whether we can determine the regularization parameter $\alpha^{(k)}$ corresponding to a given solution $\mathbf{w}^{(k)}$. Specifically, given that

$$\mathbf{w}^{(k)} = \arg \min_{\mathbf{w}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X} \mathbf{w}\|_2^2 + \alpha^{(k)} \|\mathbf{w}\|_1, \quad (\text{B.7})$$

we seek to find $\alpha^{(k)}$. For the special case $k = 0$, we refer to Appendix A. The solution $\mathbf{w}^{(k)}$ for $k > 0$ must fulfill the necessary condition for a minimum

$$\mathbf{0} \in -\frac{1}{n} \mathbf{X}^T (\mathbf{y} - \mathbf{X} \mathbf{w}^{(k)}) + \alpha^{(k)} \text{sign}(\mathbf{w}^{(k)}) = -\frac{1}{n} \mathbf{c}^{(k)} + \alpha^{(k)} \text{sign}(\mathbf{w}^{(k)}), \quad (\text{B.8})$$

where we defined

$$\text{sign}(\mathbf{w}^{(k)})_i = \begin{cases} \{-1\} & \text{if } w_i^{(k)} < 0 \\ [-1, 1] & \text{if } w_i^{(k)} = 0. \\ \{1\} & \text{if } w_i^{(k)} > 0 \end{cases} \quad (\text{B.9})$$

Thus, for a parameter $w_{i^*}^{(k)} \neq 0$ with $i^* \in \mathcal{A}^{(k)}$, we obtain

$$0 = -\frac{1}{n} c_{i^*}^{(k)} + \alpha^{(k)} \text{sign}(w_{i^*}^{(k)}), \quad (\text{B.10})$$

from which we deduce the regularization parameter

$$\alpha^{(k)} = \frac{c_{i^*}^{(k)}}{\text{sign}(w_{i^*}^{(k)})n} = \frac{\bar{c}_{\max}^{(k)}}{n}, \quad (\text{B.11})$$

where we used $\text{sign}(w_{i^*}^{(k)}) = \text{sign}(c_{i^*}^{(k)})$, see Appendix B.2.

Algorithm 5 Modified *Least angle regression* (LARS-LASSO).

Given $\mathbf{X}$ and $\mathbf{y}$
 Set $k = 0$, $\mathbf{w}^{(0)} = \mathbf{0}$, $\tilde{i} = -1$
 $\mathbf{y}_{\parallel} = \mathbf{X} [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$
 $\mathbf{c}^{(0)} = \mathbf{X}^T \mathbf{y}_{\parallel}$
 $\alpha^{(0)} = \tilde{c}_{\max}^{(0)} / n$
while $\|\mathbf{w}^{(k)}\|_0 < m - 1$ **do**
 $\mathcal{A}^{(k)} = \left\{ i^* \in \{1, \dots, m\} \setminus \{\tilde{i}\} \mid |c_{i^*}^{(k)}| = \tilde{c}_{\max}^{(k)} = \max_i |c_i^{(k)}| \right\}$, $\mathcal{A}^{\mathcal{L}(k)} = \left\{ j^* \in \{1, \dots, m\} \mid j^* \notin \mathcal{A}^{(k)} \right\}$
 $\mathbf{s}^{(k)} = \text{sign}(\mathbf{c}^{(k)})$
 $\tilde{\mathbf{X}}_{\mathcal{A}}^{(k)} = \mathbf{X}_{\mathcal{A}} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)})$
 $\mathbf{A}^{(k)} = 1 / \sqrt{\mathbf{1}_{\mathcal{A}}^T [\tilde{\mathbf{X}}_{\mathcal{A}}^{T(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}}$
 $\mathbf{u}^{(k)} = \mathbf{A}^{(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)} [\tilde{\mathbf{X}}_{\mathcal{A}}^{T(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$
 $\gamma^{(k)} \leftarrow \min_{j^* \in \mathcal{A}^{\mathcal{L}(k)}}^+ \left\{ \frac{\tilde{c}_{\max}^{(k)} - c_{j^*}^{(k)}}{\mathbf{A}^{(k)} - d_{j^*}^{(k)}}, \frac{\tilde{c}_{\max}^{(k)} + c_{j^*}^{(k)}}{\mathbf{A}^{(k)} + d_{j^*}^{(k)}} \right\}$
 $\mathbf{d}_{\mathcal{A}}^{(k)} = \mathbf{A}^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) [\tilde{\mathbf{X}}_{\mathcal{A}}^{T(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$
 $\tilde{\gamma} \leftarrow \min_{i^* \in \mathcal{A}^{(k)}}^+ \{-w_{i^*}^{(k)} / d_{i^*}^{(k)}\}$
 if $\{-w_{i^*}^{(k)} / d_{i^*}^{(k)}\} \cap \mathbb{R}^+ \neq \emptyset$ **and** $\tilde{\gamma} < \gamma^{(k)}$ **then**
 $\gamma^{(k)} \leftarrow \tilde{\gamma}$
 $\tilde{i} \leftarrow \arg \min_{i^* \in \mathcal{A}^{(k)}}^+ \{-w_{i^*}^{(k)} / d_{i^*}^{(k)}\}$
 else
 $\tilde{i} \leftarrow -1$
 end if
 $\mathbf{w}_{\mathcal{A}^{\mathcal{L}}}^{(k+1)} \leftarrow \mathbf{w}_{\mathcal{A}^{\mathcal{L}}}^{(k)}$
 $\mathbf{w}_{\mathcal{A}}^{(k+1)} \leftarrow \mathbf{w}_{\mathcal{A}}^{(k)} + \gamma^{(k)} \mathbf{A}^{(k)} \text{diag}(\mathbf{s}_{\mathcal{A}}^{(k)}) [\tilde{\mathbf{X}}_{\mathcal{A}}^{T(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)}]^{-1} \mathbf{1}_{\mathcal{A}}$
 $\mathbf{c}^{(k+1)} = \mathbf{X}^T [\mathbf{y}_{\parallel} - \mathbf{X} \mathbf{w}^{(k+1)}]$
 $\alpha^{(k+1)} = \tilde{c}_{\max}^{(k+1)} / n$
 $k \leftarrow k + 1$
end while
 $\mathbf{w}^{(k+1)} \leftarrow [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}$
 $\alpha^{(k+1)} = 0$

Appendix C. Implementation of the algorithms**C.1. CD**

We implement CD in Python using `numpy` version 2.2.2. We note that CD is also implemented in the subroutine `Lasso` in `scikit-learn` version 1.6.1.

C.2. LARS and LARS-LASSO

[Algorithm 5](#) provides a detailed description of LARS-LASSO. We implement LARS and LARS-LASSO in Python using `numpy` version 2.2.2. We note that LARS and LARS-LASSO are also implemented in the subroutine `lars_path` in `scikit-learn` version 1.6.1.

C.2.1. Evaluating the equality of numbers

An important question is how to evaluate numerically whether two numbers are equal, or equivalently how to evaluate whether a number is zero. For example, in [Algorithm 2](#) and [5](#), the active set $\mathcal{A}^{(k)}$ is determined by identifying the maximum absolute correlation and checking which feature vectors correspond to an equal correlation in magnitude. Further, when computing the L_0 -pseudo-norm $\|\mathbf{w}^{(k)}\|_0$ in [Algorithms 2](#) and [5](#), we have to determine whether a number is zero. To avoid the influence of small numerical variations, we define a tolerance value. If the absolute difference between two numbers is below the tolerance, we claim that these numbers are equal. Equivalently, if the absolute value of a number is below the tolerance, we claim that this number is zero. In the code, we choose this tolerance as 10^{-12} .

C.2.2. Early stopping

LARS and LARS-LASSO, see Algorithms 2 and 5, can exhibit unpredictable behavior as the active set increases. Particularly, solving the linear system $\left[\tilde{\mathbf{X}}_{\mathcal{A}}^{T(k)} \tilde{\mathbf{X}}_{\mathcal{A}}^{(k)}\right]^{-1} \mathbf{1}_{\mathcal{A}}$ may become ill-conditioned. Following the implementation of the subroutine `lars_path` in `scikit-learn` version 1.6.1, we stop the iterations once $\alpha = \bar{c}_{\max}/n$ is below a threshold of `np.finfo(np.float32).eps`.

C.3. ISTA

We implement ISTA and pathwise ISTA in Python using `numpy` version 2.2.2 and `PyTorch`, i.e., `torch` version 2.6.0. Specifically, we use `PyTorch` to efficiently compute the gradient of $f(\mathbf{w})$ using automatic differentiation.

References

- [1] J. Ghaboussi, J.H. Garrett, X. Wu, Knowledge-based modeling of material behavior with neural networks, *J. Eng. Mech.* 117 (1) (1991) 132–153. [https://doi.org/10.1061/\(ASCE\)0733-9399\(1991\)117:1\(132\)](https://doi.org/10.1061/(ASCE)0733-9399(1991)117:1(132))
- [2] T. Sussman, K.J. Bathe, A model of incompressible isotropic hyperelastic material behavior using spline interpolations of tension-compression test data, *Commun. Numer. Methods Eng.* 25 (1) (2009) 53–63. <https://doi.org/10.1002/cnm.1105>
- [3] N. Vlassis, R. Ma, W. Sun, Geometric deep learning for computational mechanics Part I: anisotropic hyperelasticity, (2020). [arXiv:2001.04292](https://arxiv.org/abs/2001.04292) [cs], <https://doi.org/10.1016/j.cma.2020.113299>
- [4] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based artificial neural networks for constitutive modeling, *J. Mech. Phys. Solids* 147 (2021) 104277. <https://doi.org/10.1016/j.jmps.2020.104277>
- [5] K. Linka, M. Hillgärtner, K.P. Abdolazizi, R.C. Aydin, M. Itskov, C.J. Cyron, Constitutive artificial neural networks: a fast and general approach to predictive data-driven constitutive modeling by deep learning, *J. Comput. Phys.* 429 (2021) 110010. <https://doi.org/10.1016/j.jcp.2020.110010>
- [6] C. Bonatti, D. Mohr, One for all: universal material model based on minimal state-space neural networks, *Sci. Adv.* 7 (26) (2021) eabf3658. <https://doi.org/10.1126/sciadv.abf3658>
- [7] D.K. Klein, M. Fernández, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, *J. Mech. Phys. Solids* 159 (2022) 104703. <https://doi.org/10.1016/j.jmps.2021.104703>
- [8] F. As'ad, C. Farhat, A mechanics-informed neural network framework for data-driven nonlinear viscoelasticity (2022). Publisher: Unpublished, <https://doi.org/10.13140/RG.2.2.21694.36168>
- [9] J.N. Fuhg, M. Marino, N. Bouklas, Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks, *Comput. Methods Appl. Mech. Eng.* 388 (2022) 114217. <https://doi.org/10.1016/j.cma.2021.114217>
- [10] V. Tac, F. Sahli Costabal, A.B. Tepole, Data-driven tissue mechanics with polyconvex neural ordinary differential equations, *Comput. Methods Appl. Mech. Eng.* 398 (2022) 115248. <https://doi.org/10.1016/j.cma.2022.115248>
- [11] P. Thakolkaran, A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, S. Kumar, NN-EUCLID: deep-learning hyperelasticity without stress data, *J. Mech. Phys. Solids* 169 (2022) 105076. <https://doi.org/10.1016/j.jmps.2022.105076>
- [12] K.A. Kalina, L. Linden, J. Brummund, P. Metsch, M. Kästner, Automated constitutive modeling of isotropic hyperelasticity based on artificial neural networks, *Comput. Mech.* 69 (1) (2022) 213–232. <https://doi.org/10.1007/s00466-021-02090-6>
- [13] M. Rosenkranz, K.A. Kalina, J. Brummund, M. Kästner, A comparative study on different neural network architectures to model inelasticity, *Int. J. Numer. Methods Eng.* 124 (21) (2023) 4802–4840. <https://doi.org/10.1002/nme.7319>
- [14] A. Benady, E. Baranger, L. Chamoin, Unsupervised learning of history-dependent constitutive material laws with thermodynamically-consistent neural networks in the modified constitutive relation error framework, *Comput. Methods Appl. Mech. Eng.* 425 (2024) 116967. <https://doi.org/10.1016/j.cma.2024.116967>
- [15] M. Flaschel, P. Steinmann, L. De Lorenzis, E. Kuhl, Convex neural networks learn generalized standard material models, *J. Mech. Phys. Solids* 200 (2025) 106103. <https://doi.org/10.1016/j.jmps.2025.106103>
- [16] J. Bleyer, Learning elastoplasticity with implicit layers (2025). <https://doi.org/10.2139/ssrn.5210734>
- [17] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Eng.* 304 (2016) 81–101. <https://doi.org/10.1016/j.cma.2016.02.001>
- [18] R. Ibañez, D. Borzacchiello, J.V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladeveze, F. Chinesta, Data-driven non-linear elasticity: constitutive manifold construction and problem discretization, *Comput. Mech.* 60 (5) (2017) 813–826. <https://doi.org/10.1007/s00466-017-1440-1>
- [19] M. Schoenauer, M. Sebag, F. Jouve, B. Lamy, H. Maitournam, Evolutionary identification of macro-mechanical models, In: *Advances in Genetic Programming II*, ed. by P.J. Angeline and K.E. Kinneer Jr., MIT Press, Cambridge, MA, (1996) 467–488.
- [20] A. Ratle, M. Sebag, Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics, *Appl. Soft Comput.* 1 (1) (2001) 105–118. [https://doi.org/10.1016/S1568-4946\(01\)00009-6](https://doi.org/10.1016/S1568-4946(01)00009-6)
- [21] D. Versino, A. Tonda, C.A. Bronkhorst, Data driven modeling of plastic deformation, *Comput. Methods Appl. Mech. Eng.* 318 (2017) 981–1004. <https://doi.org/10.1016/j.cma.2017.02.016>
- [22] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, *Comput. Methods Appl. Mech. Eng.* 381 (2021) 113852. <https://doi.org/10.1016/j.cma.2021.113852>
- [23] G.F. Bomarito, T.S. Townsend, K.M. Stewart, K.V. Esham, J.M. Emery, J.D. Hochhalter, Development of interpretable, data-driven plasticity models with symbolic regression, *Comput. Struct.* 252 (2021) 106557. <https://doi.org/10.1016/j.compstruc.2021.106557>
- [24] H. Park, M. Cho, Multiscale constitutive model using data-driven yield function, *Compos. B Eng.* 216 (2021) 108831. <https://doi.org/10.1016/j.compositesb.2021.108831>
- [25] Z. Wang, J.B. Estrada, E.M. Arruda, K. Garikipati, Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification, *J. Mech. Phys. Solids* 153 (2021) 104474. <https://doi.org/10.1016/j.jmps.2021.104474>
- [26] M. Wang, C. Chen, W. Liu, Establish algebraic data-driven constitutive models for elastic solids with a tensorial sparse symbolic regression method and a hybrid feature selection technique, *J. Mech. Phys. Solids* 159 (2022) 104742.
- [27] R. Abusalamov, M. Hillgärtner, M. Itskov, Automatic generation of interpretable hyperelastic material models by symbolic regression, *Int. J. Numer. Methods Eng.* (2023) nme.7203. <https://doi.org/10.1002/nme.7203>
- [28] K. Linka, E. Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, *Comput. Methods Appl. Mech. Eng.* 403 (2023) 115731. <https://doi.org/10.1016/j.cma.2022.115731>
- [29] K.A. Meyer, F. Ekre, Thermodynamically consistent neural network plasticity modeling and discovery of evolution laws, 2023. <https://doi.org/10.31224/2961>
- [30] J.N. Fuhg, R.E. Jones, N. Bouklas, Extreme sparsification of physics-augmented neural networks for interpretable model discovery in mechanics, *Comput. Methods Appl. Mech. Eng.* 426 (2024) 116973. <https://doi.org/10.1016/j.cma.2024.116973>
- [31] J. Hou, X. Chen, T. Wu, E. Kuhl, X. Wang, Automated data-driven discovery of material models based on symbolic regression: a case study on human brain cortex, *Acta Biomater.* 188 (2024) 276–296.
- [32] B. Bahmani, W. Sun, Physics-constrained symbolic model discovery for polyconvex incompressible hyperelastic materials, *Int. J. Numer. Methods Eng.* 125 (15) (2024) e7473. <https://doi.org/10.1002/nme.7473>

- [33] G. Kissas, S. Mishra, E. Chatzi, L. De Lorenzis, The language of hyperelastic materials, *Comput. Methods Appl. Mech. Eng.* 428 (2024) 117053. <https://doi.org/10.1016/j.cma.2024.117053>
- [34] P. Thakolkaran, Y. Guo, S. Saini, M. Peirlinck, B. Alheit, S. Kumar, Can KAN CANs? input-convex Kolmogorov-Arnold networks (KANs) as hyperelastic constitutive artificial neural networks (CANs), 2025. <https://doi.org/10.48550/arXiv.2503.05617>
- [35] K.P. Abdolazizi, R.C. Aydin, C.J. Cyron, K. Linka, Constitutive Kolmogorov-Arnold networks (CKANs): combining accuracy and interpretability in data-driven material modeling, 2025. <https://doi.org/10.48550/arXiv.2502.05682>
- [36] J.N. Fuhg, G. Anantha Padmanabha, N. Bouklas, B. Bahmani, W. Sun, N.N. Vlassis, M. Flaschel, P. Carrara, L. De Lorenzis, A review on data-driven constitutive laws for solids, *Arch. Comput. Methods Eng.* (2024). <https://doi.org/10.1007/s11831-024-10196-2>
- [37] M. Peirlinck, K. Linka, J.A. Hurtado, G.A. Holzapfel, E. Kuhl, Democratizing biomedical simulation through automated model discovery and a universal material subroutine, *Comput. Mech.* (2024). <https://doi.org/10.1007/s00466-024-02515-y>
- [38] M. Flaschel, S. Kumar, L. De Lorenzis, Automated discovery of generalized standard material models with EUCLID, *Comput. Methods Appl. Mech. Eng.* 405 (2023) 115867. <https://doi.org/10.1016/j.cma.2022.115867>
- [39] F. Santosa, W.W. Symes, Linear inversion of band-limited reflection seismograms, *SIAM J. Sci. Stat. Comput.* 7 (4) (1986) 1307–1330. <https://doi.org/10.1137/0907087>
- [40] I.E. Frank, J.H. Friedman, A statistical view of some chemometrics regression tools, *Technometrics* 35 (2) (1993) 109–135. <https://doi.org/10.1080/00401706.1993.10485033>
- [41] R. Tibshirani, Regression shrinkage and selection via the Lasso, *J. R. Stat. Soc. B* 58 (1) (1996) 267–288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- [42] W. Fu, Penalized regressions: the bridge versus the Lasso (1998).
- [43] M. Osborne, B. Presnell, B.A. Turlach, A new approach to variable selection in least squares problems, *IMA J. Numer. Anal.* 20 (3) (2000) 389–403. <https://doi.org/10.1093/imanum/20.3.389>
- [44] M.R. Osborne, B. Presnell, B.A. Turlach, On the LASSO and its dual, *J. Comput. Graph. Stat.* 9 (2000) 319–337.
- [45] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, *Ann. Stat.* 32 (2) (2004) 407–499.
- [46] I. Daubechies, M. Deffrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pure Appl. Math.* 57 (11) (2004) 1413–1457. <https://doi.org/10.1002/cpa.20042>
- [47] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, *J. R. Stat. Soc. B* 67 (2) (2005) 301–320. <https://doi.org/10.1111/j.1467-9868.2005.00503.x>
- [48] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, Pathwise coordinate optimization, *Ann. Appl. Stat.* 1 (2) (2007). <https://doi.org/10.1214/07-AOAS131>
- [49] S.J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale ℓ_1 -regularized least squares, *IEEE J. Sel. Top. Signal Process.* 1 (4) (2007) 606–617. <https://doi.org/10.1109/JSTSP.2007.910971>
- [50] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci.* 113 (15) (2016) 3932–3937. <https://doi.org/10.1073/pnas.1517384113>
- [51] K. Linka, S.P. Sarah R., E. Kuhl, Automated model discovery for human brain using constitutive artificial neural networks, *Acta Biomater.* 160 (2023) 134–151. <https://doi.org/10.1016/j.actbio.2023.01.055>
- [52] S.P. Skyler R., D. Rajasekharan, E.C. Darwin, K. Linka, M.E. Levenston, E. Kuhl, Discovering the mechanics of artificial and real meat, *Comput. Methods Appl. Mech. Eng.* 415 (2023) 116236. <https://doi.org/10.1016/j.cma.2023.116236>
- [53] S.P. Sarah R., K. Linka, E. Kuhl, Principal-stretch-based constitutive neural networks autonomously discover a subclass of Ogden models for human brain tissue, *Brain Multiphysics* 4 (2023) 100066. <https://doi.org/10.1016/j.brain.2023.100066>
- [54] M. Flaschel, H. Yu, N. Reiter, J. Hinrichsen, S. Budday, P. Steinmann, S. Kumar, L. De Lorenzis, Automated discovery of interpretable hyperelastic material models for human brain tissue with EUCLID, *J. Mech. Phys. Solids* 180 (2023) 105404. <https://doi.org/10.1016/j.jmps.2023.105404>
- [55] K. Linka, E. Kuhl, Best-in-class modeling: a novel strategy to discover constitutive models for soft matter systems, *Extreme Mech. Lett.* 70 (2024) 102181. <https://doi.org/10.1016/j.eml.2024.102181>
- [56] H. Moon, D. Park, H. Cho, H.K. Noh, Physics-informed neural network-based discovery of hyperelastic constitutive models from extremely scarce data, (2025). [arXiv:2504.19494](https://arxiv.org/abs/2504.19494)
- [57] E. Marino, M. Flaschel, S. Kumar, L. De Lorenzis, Automated identification of linear viscoelastic constitutive laws with EUCLID, *Mech. Mater.* 181 (2023) 104643. <https://doi.org/10.1016/j.mechmat.2023.104643>
- [58] M. Flaschel, S. Kumar, L. De Lorenzis, Discovering plasticity models without stress data, *Npj Comput. Mater.* 8 (1) (2022) 91. <https://doi.org/10.1038/s41524-022-00752-4>
- [59] H. Xu, M. Flaschel, L. De Lorenzis, Discovering non-associated pressure-sensitive plasticity models with EUCLID, *Adv. Model. Simul. Eng. Sci.* 12 (1) (2025) 1. <https://doi.org/10.1186/s40323-024-00281-3>
- [60] H. Holthusen, T. Brepols, K. Linka, E. Kuhl, Automated model discovery for tensional homeostasis: constitutive machine learning in growth and remodeling, 2024. [arXiv:2410.13645](https://arxiv.org/abs/2410.13645) [cs]
- [61] Y. Kanno, *Nonsmooth Mechanics and Convex Optimization* (1st ed.), CRC Press, 2011.
- [62] J. Bleyer, Applications of conic programming in non-smooth mechanics, *J. Optim. Theory Appl.* 202 (1) (2024) 340–372. <https://doi.org/10.1007/s10957-022-02105-z>
- [63] J. Bleyer, *Variational principles in nonlinear mechanics using convex optimization and automated numerical tools*, 2024.
- [64] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, Springer New York, New York, NY, 2009. <https://doi.org/10.1007/978-0-387-84858-7>
- [65] N. Parikh, S. Boyd, Proximal algorithms, *Found. Trends Optim.* 1 (2013) 123–231.
- [66] A. Beck, *First-Order Methods in Optimization*, MOS-SIAM Series on Optimization, 2017.
- [67] J. Urrea-Quintero, D. Anton, L. De Lorenzis, H. Wessels, Automated constitutive model discovery by pairing sparse regression algorithms with model selection criteria, *Comput. Methods Appl. Mech. Eng.* 449 (2026) 118551. <https://doi.org/10.1016/j.cma.2025.118551>
- [68] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, *J. Stat. Softw.* 33 (1) (2010). <https://doi.org/10.18637/jss.v033.i01>
- [69] J. Yang, T. Hastie, A fast and scalable pathwise-solver for group Lasso and elastic net penalized regression via block-coordinate descent, 2024. <https://doi.org/10.48550/arXiv.2405.08631>
- [70] J. Yang, T. Hastie, A fast coordinate descent method for high-dimensional non-negative least squares using a unified sparse regression framework, 2024. <https://doi.org/10.48550/arXiv.2410.03014>
- [71] G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor, *An Introduction to Statistical Learning with Applications in Python*, 2023.
- [72] T. Hesterberg, N.H. Choi, L. Meier, C. Fraley, Least angle and l1 penalized regression: a review, *Stat. Surv.* 2 (2008) 61–93.
- [73] J. Fan, J. Lv, A selective overview of variable selection in high dimensional feature space, *Stat Sin.* 20 (1) (2010) 101–148.
- [74] D. Vidaurre, C. Bielza, P. Larrañaga, A survey of L1 regression, *Int. Stat. Rev.* 81 (2013) 361–387.
- [75] H. Zou, L. Xue, A selective overview of sparse principal component analysis, *Proc. IEEE* 106 (8) (2018).
- [76] N. Lüthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: literature survey and benchmark, *SIAM/ASA J. Uncertain. Quantif.* 9 (2) (2021) 593–649. <https://doi.org/10.1137/20M1315774>
- [77] J. Koza, Genetic programming as a means for programming computers by natural selection, *Stat. Comput.* 4 (2) (1994). <https://doi.org/10.1007/BF00175355>
- [78] D.P. Seanson, D.E. Leahy, M.J. Willis, GPTIPS: an open source genetic programming toolbox for multigene symbolic regression, *Proc. Int. MultiConf. Eng. Comput. Sci.* 1 (2010) 77–80.
- [79] R. Dubčáková, Eureka: software review, *Genet. Program. Evolvable Mach.* 12 (2011) 173–178.

- [80] S.-M. Udrescu, M. Tegmark, AI Feynman: a physics-inspired method for symbolic regression, *Sci. Adv.* 6 (16) (2020) eaay2631. <https://doi.org/10.1126/sciadv.aay2631>
- [81] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, *Science* 324 (5923) (2009) 81–85. <https://doi.org/10.1126/science.1165893>
- [82] E. Kablman, A.H. Kolody, J. Kronsteiner, M. Kommenda, G. Kronberger, Application of symbolic regression for constitutive modeling of plastic deformation, *Appl. Eng. Sci.* 6 (2021) 100052. <https://doi.org/10.1016/j.applsc.2021.100052>
- [83] J.A. McCulloch, S.P. Skyler R., K. Linka, E. Kuhl, On sparse regression, L_p -regularization, and automated model discovery, *Int. J. Numer. Methods Eng.* 125 (14) (2024) e7481. <https://doi.org/10.1002/nme.7481>
- [84] R.T. Rockafellar, *Convex Analysis*, Number 28 in Princeton mathematical series, Princeton University Press, Princeton, N.J., 1970.
- [85] S.P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge New York Melbourne New Delhi Singapore, version 29 edition, 2004.
- [86] G. Marckmann, E. Verron, Comparison of hyperelastic models for rubber-like materials, *Rubber Chem. Technol.* 79 (5) (2006) 835–858. <https://doi.org/10.5254/1.3547969>
- [87] G. Chagnon, M. Rebouah, D. Favier, Hyperelastic energy densities for soft biological tissues: a review, *J. Elast.* 120 (2) (2015) 129–160. <https://doi.org/10.1007/s10659-014-9508-z>
- [88] H. Dal, K. Açıkgöz, Y. Badienia, On the performance of isotropic hyperelastic constitutive models for rubber-like materials: a state of the art review, *Appl. Mech. Rev.* 73 (2) (2021) 020802. <https://doi.org/10.1115/1.4050978>
- [89] G.A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, Wiley, Chichester; New York, 2000.
- [90] R.S. Rivlin, Torsion of a rubber cylinder, *J. Appl. Phys.* 18 (1947) 444–449.
- [91] Rivlin, Large elastic deformations of isotropic materials. I. fundamental concepts (1950) 32.
- [92] Rivlin, Large elastic deformations of isotropic materials VII. experiments on the deformation of rubber, *Philos. Trans. R. Soc. London Ser. A Math. Phys. Sci.* 243 (865) (1951) 251–288. <https://doi.org/10.1098/rsta.1951.0004>
- [93] S. Hartmann, Parameter estimation of hyperelasticity relations of generalized polynomial-type with constraint conditions, *Int. J. Solids Struct.* 38 (44–45) (2001) 7999–8018. [https://doi.org/10.1016/S0020-7683\(01\)00018-X](https://doi.org/10.1016/S0020-7683(01)00018-X)
- [94] R.W. Ogden, Large deformation isotropic elasticity—on the correlation of theory and experiment for incompressible rubberlike solids, *Proc. R. Soc. London Ser. A Math. Phys. Sci.* 326 (1972) 565–584.
- [95] M. Flaschel, Automated Discovery of Material Models in Continuum Solid Mechanics, Ph.D. thesis, ETH Zurich, 2023. <https://doi.org/10.3929/ETHZ-B-000602750>
- [96] M. Flaschel, T. Hastie, E. Kuhl, Supplementary software for “non-smooth optimization meets automated material model discovery”, Zenodo (2025). <https://doi.org/10.5281/zenodo.15848305>