Elasto-plasticity with convex model-data-driven yield functions

Jan Niklas Fuhg, Amélie Fau, Nikolaos Bouklas, Michele Marino

ELASTO-PLASTICITY WITH CONVEX MODEL-DATA-DRIVEN YIELD FUNCTIONS

A PREPRINT

Jan N. Fuhg
Cornell University
Ithaca, NY, 14853, United States

Amélie Fau
Université Paris-Saclay, CentraleSupélec,
ENS Paris-Saclay, CNRS,
LMPS - Laboratoire de Mécanique Paris-Saclay,
91190, Gif-sur-Yvette, France

Nikolaos Bouklas
Cornell University
Ithaca, NY, 14853, United States

Michele Marino
University of Rome Tor Vergata,
Via del Politecnico 1, 00133 Rome, Italy.

March 21, 2022

ABSTRACT

The formulation of history-dependent material laws has been a significant challenge in solid mechanics for over a century. Recently, data-driven techniques have generated accurate and reliable surrogates for elasto-plastic constitutive laws. However, most of these methods are deeply rooted in the big data domain and fail when only a few physically obtained experimental data points are available. To combat this, we propose a plasticity formulation with model-data-driven yield functions that is designed to work in the small data regime. This is done by locally improving a phenomenological yield function (model component) with a data-driven correction term (data component) which only utilizes uniaxial and biaxial experimental data describing the shape of the initial yield surface. This allows seamless merging of conventional material models with their data-driven counterparts enabling the derivation of hybrid models that significantly improve the accuracy and robustness of traditional approaches. In order to obtain convex yield functions in this framework, it is sufficient for the data-driven correction to be convex. Strategies based on convex extensions to Support Vector Regression (SVR), Gaussian Process Regression (GPR), and Neural Networks (NN) are analysed. The proposed approach is tested on synthetic data of anisotropic yield functions commonly used for rolled metal sheets.

Keywords Data-driven elasto-plasticity · Physics-informed machine learning · Machine learning · Convex yield function · Anisotropic yield function

1 Introduction

Machine learning and data-driven techniques are currently one of the main drivers of advancement in engineering science. They allow to make efficient use of available data while allowing to automate training processes aiding quick prototyping. Hence, machine learning solutions have been employed in various fields to build surrogates or reduced-order models for complex physical mappings [1, 2, 3]. On the other hand, in computational physics, they are also extensively used as direct solvers for deterministic [4, 5] or uncertain [6, 7] partial differential equations.

Recently, machine learning approaches have also seen an increased interest in material modeling. Data-driven constitutive modeling has been used to create surrogates of nonlinear path-independent material laws such as hyperelasticity [8, 9, 10]. Alternatively, hybrid models have been proposed where known traditional phenomenological models are locally improved with machine-learning-based correction terms when data is available [11, 12, 13]. In Ref. [12], the term model-data-driven material modeling for these types of hybrid constitutive laws has been established. In

contrast to path-independent material behavior where one-to-one mappings between the strain and stress significantly simplify material modeling, path-dependent closure models rely on time- and/or history-evolution of the material. In terms of plasticity, different theoretical frameworks for phenomenological approaches, such as hypo-plasticity [14] or elasto-plasticity [15], have been developed over the years. Elasto-plasticity, for example, distinguishes between elastic and inelastic constitutive responses by employing yield surfaces. On the other hand, hypo-plastic models make no distinction between elastic and plastic strains and do not use yield functions to characterize the onset of yielding. The divide between these two classical theories in phenomenological elasto-plastic modeling can also be found in their data-driven counterparts.

The earliest machine-learning-based approaches for plasticity (see, e.g., [16, 17]) treat the material law as a problem defined by sequential stress-strain data, which is in line with the ideas from hypo-plasticity. This idea has also been applied to a majority of the current works. For example, [18] use recurrent neural networks to successfully map between history-dependent stress-strain data points where the internal feedback connections are used to find embedded internal variables implicitly. Other approaches, such as [19] utilize neural networks combined with an experimentally obtainable internal history variable to generate a data-driven plasticity model in two- and three dimensions. However, these approaches suffer from problems with interpretability since it is unclear how the model behaves on unseen data, i.e. data outside the training range. Secondly, efficient sampling schemes are needed that generate representative data that allows to extrapolate to other load cases. This last point has not yet been addressed in a sufficient manner.

Recently, [20] propose an approach that is more in line with elasto-plastic theory. Here, the material law is constructed out of modular components, similar to what one would encounter in an analytical formulation, where the elastic and plastic contributions (i.e., yield function) can be trained with separate but complementing predictive tools. This allows for better interpretability and reliability than traditional black-box surrogates. With this idea in mind, [21] introduce a framework to generate texture-dependent yield functions with neural networks which are strictly convex but can also embed microstructural information. Other machine learning-based methods for yield functions have been proposed. A correction term to a yield function was applied in [22]. [23] explores Support Vector Regression as a tool to determine anisotropic yield functions. Symbolic regression is used in [24] and [25] to find interpretable expressions for isotropic yield functions based on data. [26] introduce an approach that fits symbolic regression models to synthetic Digital Image Correlation data. However, all of the discussed models are part of a big-data mindset where a significant amount of data from computer simulations or from full-field measurements [27] is assumed to be available. Thus, these approaches can not be directly applied using only experimental data that are commonly available from uniaxial and biaxial test as these would probe only a small subset of the stress space.

In this work we propose a hybrid model-data-driven approach that can locally improve phenomenological yield functions where experimental data is available. The framework is developed with data availability, and the idea of data-poor modeling in mind. As a proof-of-concept we propose and study the framework on perfect plasticity. This restriction is chosen deliberately since experimental data describing the shape of the yield function is (typically) only available for the initial yield surface. In future works, we aim to extend it to include hardening.

Typically, it can be seen as a physical requirement that the yield function is convex with regard to its stress-dependent arguments. Hence, in this work, special attention is given to ensure the convexity of the resulting model-data-driven yield function. We recognize that the convexity of the correction term guarantees convexity of the final yield function (assuming that the model component in the hybrid formulation is convex). Hence, the convex extensions to three commonly applied machine learning methods are introduced. These include Input Convex Support Vector Regression, Input Convex Gaussian Process Regression, and Input Convex Neural Networks. The framework is tested for modeling rolled metal sheets which are typically characterized by anisotropic yield. The organization of the paper is as follows. The concept of elasto-plasticity with model-data-driven yield functions in the context of the studied use-case is thoroughly explained in Section 2. The considered convexity constrained machine learning techniques are reviewed in Section 3. The numerical implementation of the framework is detailed in Section 4. Results of model-data-driven elasto-plasticity utilized on rolled sheet metals and compared to pure-data-driven approaches are highlighted in Section 5. The consequences of these findings are discussed in Section 6. The paper is concluded in Section 7.

2 Materials and Methods

This paper deals with the characterization and modeling of yield functions for metallic materials. The focus is in particular on rolled metal sheets, where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ denote the rolling, transverse, and normal directions (RD, TD, and ND), respectively. These types of materials are generally subjected to a plane stress condition in the $(\mathbf{x}, \mathbf{y})$ plane. With obvious notation, σ_x, σ_y and σ_{xy} respectively represent normal components along the RD and TD, as well as the shear component in the RD-TD plane, of the Cauchy stress tensor $\boldsymbol{\sigma}$.

The total strain $\boldsymbol{\varepsilon}$ is additively split into an elastic $\boldsymbol{\varepsilon}_e$ and plastic $\boldsymbol{\varepsilon}_p$ part:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_e + \boldsymbol{\varepsilon}_p \,. \tag{1}$$

and the the Cauchy stress is assumed to be linearly dependent on the elastic strain

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}_e = \mathbb{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_p) \tag{2}$$

where $\mathbb{C}$ is the elasticity tensor.

2.1 Model-data-driven approach

In the proposed model-data-driven approach, plastic flow is described by the following yield function:

$$f(\boldsymbol{\sigma}) = f_{mod}(\boldsymbol{\sigma}) + f_{rem}(\boldsymbol{\sigma}) \leq 0 \,, \tag{3}$$

where f_{mod} represents a given model component and f_{rem} is the remainder between the exact and model-based yield functions.

In the present work, no hardening effects are accounted. These can be included by adding an explicit dependence on $\boldsymbol{\varepsilon}_p$. We restrict ourselves to perfect plasticity, since, data describing the shape of the yield surface is typically only available for the initial surface.

An associative evolution law for the plastic strain is considered, that is

$$\dot{\boldsymbol{\varepsilon}}_p = \begin{cases} \dot{\lambda}\left[\mathbf{R}_\sigma^{mod}(\boldsymbol{\sigma}) + \mathbf{R}_\sigma^{rem}(\boldsymbol{\sigma})\right] & \text{if } f(\boldsymbol{\sigma}) = 0 \\ \mathbf{0} & \text{if } f(\boldsymbol{\sigma}) < 0 \end{cases} , \tag{4}$$

where

$$\mathbf{R}_\sigma^{mod} = \frac{\partial f_{mod}}{\partial \boldsymbol{\sigma}} \,, \qquad \mathbf{R}_\sigma^{rem} = \frac{\partial f_{rem}}{\partial \boldsymbol{\sigma}} \,, \tag{5}$$

and λ is a scalar (positive) plastic multiplier, whose rate $\dot{\lambda}$ is required to be positive, i.e. $\dot{\lambda} \geq 0$, and different from zero only in presence of plastic flow.

Traditionally, yield functions models are established from uniaxial and biaxial experimental data [28]. These two are briefly recapitulated in the following.

2.2 Experimental data

Addressing uniaxial tension along direction $\mathbf{t}(\theta) = \mathbf{x}\cos\theta + \mathbf{y}\sin\theta$ (at angle θ with respect to the $\mathbf{x}$-axis in the $(\mathbf{x}, \mathbf{y})$ plane), the applied stress tensor results in $\boldsymbol{\sigma}_\theta^{uni}(s) = s(\mathbf{t}(\theta) \otimes \mathbf{t}(\theta))$. The corresponding yield limit, that is stress s such that $f(\boldsymbol{\sigma}_\theta^{uni}(s)) = 0$, is denoted by σ_θ^u. Another loading case study relevant from the experimental point of view is the application of a biaxial stress state in the RD-TD plane with $\tan\beta$ being the stress ratio between TD and RD. In this case, the stress tensor reads $\boldsymbol{\sigma}_\beta^{bi}(s) = s(\mathbf{x} \otimes \mathbf{x}) + s\tan\beta(\mathbf{y} \otimes \mathbf{y})$[1], and the corresponding biaxial yield limit, i.e. stress s such that $f(\boldsymbol{\sigma}_\beta^{bi}(s)) = 0$, is denoted by σ_β^b. Superscripts T and C are employed for indicating tensile (i.e., $\sigma_\theta^u, \sigma_\beta^b > 0$) or compressive (i.e., $\sigma_\theta^u, \sigma_\beta^b < 0$) stress states, respectively.

From the kinematic point of view, plastic flow can be characterized by experimental observations on Lankford coefficients r_θ^u [28, 29]. These coefficients are obtained from uniaxial tension tests along $\mathbf{t}(\theta)$. Defining $\mathbf{t}_\perp(\theta) = -\sin\theta\mathbf{x} + \cos\theta\mathbf{y}$ as the orthogonal direction to $\mathbf{t}(\theta)$, Lankford coefficients $r_\theta^u = -d_\perp^p/d_z^p$ represent the ratio between the in-plane orthogonal $d_\perp^p = \dot{\boldsymbol{\varepsilon}}_p : (\mathbf{t}_\perp \otimes \mathbf{t}_\perp)$ and through-thickness $d_z^p = \dot{\boldsymbol{\varepsilon}}_p : (\mathbf{z} \otimes \mathbf{z})$ plastic strain increments, reading

$$r_\theta^u(\sigma_\theta^u) = -\frac{d_x^p \sin^2\theta - d_{xy}^p \sin(2\theta) + d_y^p \cos^2\theta}{d_x^p + d_y^p}\bigg|_{\boldsymbol{\sigma} = \sigma_\theta^u(\mathbf{t}\otimes\mathbf{t})} \,, \tag{6}$$

with

$$d_x^p = \dot{\boldsymbol{\varepsilon}}_p : (\mathbf{x} \otimes \mathbf{x}) = \dot{\lambda}\frac{\partial f}{\partial \sigma_x} \,, \tag{7}$$

$$d_y^p = \dot{\boldsymbol{\varepsilon}}_p : (\mathbf{y} \otimes \mathbf{y}) = \dot{\lambda}\frac{\partial f}{\partial \sigma_y} \,, \tag{8}$$

$$d_{xy}^p = \dot{\boldsymbol{\varepsilon}}_p : (\mathbf{x} \otimes \mathbf{y}) = \dot{\lambda}\frac{\partial f}{\partial \sigma_{xy}} \,. \tag{9}$$

[1] In the special case $\beta = 90°$, it results in $\boldsymbol{\sigma}_{90}^{bi}(s) = s(\mathbf{y} \otimes \mathbf{y}) = \boldsymbol{\sigma}_{90}^{uni}(s)$.

2.3 Model component of the yield function

A general expression of f_{mod}, adopted in what follows, is an invariant-based representation of the form [30]

$$f_{mod}(\boldsymbol{\sigma}) = 3J_2^0(\boldsymbol{\sigma}) - \bar{\sigma}^2\,, \tag{10}$$

where J_2^0 is the second generalized orthotropic invariant, reading

$$\begin{aligned} J_2^0(\boldsymbol{\sigma}) =& \frac{a_1}{6}(\sigma_x-\sigma_y)^2 + \frac{a_2}{6}(\sigma_y-\sigma_z)^2 + \frac{a_3}{6}(\sigma_x-\sigma_z)^2 + \\ &+ a_4\sigma_{xy}^2 + a_5\sigma_{xz}^2 + a_6\sigma_{yz}^2\,, \end{aligned} \tag{11}$$

with $a_1, \ldots, a_6$ some material constants.

We use this specific model component since according to [30] it represents the "most widely used orthotropic yield criterion for describing the directionality in plastic properties (yield stresses, strains) of sheets and plates".

It has been shown that $J_2^0(\boldsymbol{\sigma})$ is pressure insensitive and so it depends only on the deviatoric stress $\boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - p\mathbf{I}$ with $p = \text{Tr}(\boldsymbol{\sigma})/3$, and reduces to its isotropic counterpart $J_2(\boldsymbol{\sigma}) = \text{Tr}(\boldsymbol{\sigma}_{dev}^2)/2$ for $a_1 = \ldots = a_6 = 1$ (see, e.g. [31]). Besides, it has been proven that f_{mod} in eq. (10) corresponds to the orthotropic generalization of the von Mises Isotropic Criterion [32]. In the literature, eq. (10) is occasionally reformulated as:

$$\begin{aligned} f_{mod}(\boldsymbol{\sigma}) =& F(\sigma_y-\sigma_z)^2 + G(\sigma_z-\sigma_x)^2 + H(\sigma_x-\sigma_y)^2 + \\ &+ 2L\sigma_{yz}^2 + 2M\sigma_{xz}^2 + 2N\sigma_{xy}^2 - \bar{\sigma}^2\,, \end{aligned} \tag{12}$$

where F, G, H, L, M and N are material constants (univocally related to the constants $a_1, \ldots, a_6$). Under plane stress conditions ($\sigma_z = \sigma_{xz} = \sigma_{yz} = 0$) only 4 parameters ($F$, G, H and N) have an impact which can be experimentally characterized by means of 4 material tests, like for instance:

- RD-uniaxial tensile traction, providing σ_0^{uT};
- TD-uniaxial tensile traction, providing σ_{90}^{uT};
- uniaxial tensile traction at $\theta = 45°$, providing σ_{45}^{uT};
- equi-biaxial tensile traction in the RD-TD plane, with $\sigma_x = \sigma_y > 0$ and $\sigma_{xy} = 0$, providing σ_{45}^{bT}.

From these experimental values, the yield function in eq. (12) can be straightforwardly calibrated via:

$$2F = (\bar{\sigma}/\sigma_{90}^{uT})^2 + (\bar{\sigma}/\sigma_{45}^{bT})^2 - (\bar{\sigma}/\sigma_0^{uT})^2\,, \tag{13}$$

$$2G = (\bar{\sigma}/\sigma_0^{uT})^2 + (\bar{\sigma}/\sigma_{45}^{bT})^2 - (\bar{\sigma}/\sigma_{90}^{uT})^2\,, \tag{14}$$

$$2H = (\bar{\sigma}/\sigma_0^{uT})^2 + (\bar{\sigma}/\sigma_{90}^{uT})^2 - (\bar{\sigma}/\sigma_{45}^{bT})^2\,, \tag{15}$$

$$2N = 4(\bar{\sigma}/\sigma_{45}^{uT})^2 - (\bar{\sigma}/\sigma_{45}^{bT})^2\,, \tag{16}$$

with $\bar{\sigma} = \sigma_0^{uT}$. Alternative strategies based for instance on Lankford coefficients r_θ^u are available, for which the interested reader is referred to, e.g., [30].

2.3.1 Limitations of model component f_{mod}

The modeling component of the yield function $f_{mod}(\boldsymbol{\sigma})$ in eq. (12) can be effectively applied in a wide range of applications, being, in fact, the most widely used orthotropic yield criterion for describing the directionality in plastic properties of sheets and plates [30]. However, the orthotropic von Mises criterion is endowed by intrinsic limitations given by its quadratic form and the relatively low number of available parameters (4 in plane stress), [33]. Therefore, some typical experimental observations in metals cannot be accounted for, like:

- the fitting of additional uniaxial or biaxial yield stresses from the ones used for calibration, see e.g. Eqs. (13)-(16);
- the simultaneous fitting of uniaxial yield stress values σ_θ^u and Lankford coefficients r_θ^u, since the calibration can be based either on the former or the latter [30];
- typical tension-compression asymmetries experienced in material strength e.g. $\sigma_0^{uT} \neq \sigma_0^{uC}$.

To better explain these aspects, an exemplary case study is presented in Fig. 1. Here, parameters F, G, H and N have been calibrated using known data values at σ_0^{uT}, σ_{45}^{uT}, σ_{90}^{uT} and σ_{45}^{bT}. It can be seen that the calibration data is perfectly

represented, but additional experimental tests cannot be incorporated, resulting in significant inaccuracies due to the adopted yield function $f_{mod}(\boldsymbol{\sigma})$. The effect of these inaccuracies on the obtained constitutive responses are shown in Fig. 2 in terms of a strain-driven uniaxial traction test along RD. We remark that, in order to obtain a rigorous comparison between the predicted and true yield surfaces in the forthcoming numerical results, the calibration data in this work is generated by means of an analytical orthotropic yield criterion by Cazacu and Barlat [33], presented in detail in A.

The traditional approach would be to seek a refined formulation with more parameters. However, the fitting capabilities of the generalized expression might anyway be too low due to intrinsic limitations of the chosen mathematical relationship, and the modeler's choices on the adopted relationship highly affect the outcome. Moreover, the number of parameters is finite, and so will be the data that can be reproduced exactly via a proper calibration step. When more data points than parameters are available, then the parameters are generally calibrated by minimizing an error function, the form and weight factors of which introduce another impact due to decisions of the modeler.

Accordingly, in order to ensure a general accurate description of available information, the authors here propose to introduce a surrogate model for the remainder term f_{rem} in eq. (3), built by machine learning. Hence, the remainder term represents the data-driven component in the model-data-driven elasto-plastic approach.

2.4 Data-driven component

To introduce the data-driven part of our framework, first we discuss the type of data that could potentially be available from experiments. In general, the data is given as discrete values of the remainder term at given stress states. Hence, our data represents the (negative) effective value of the model yield function at the experimentally measured stress positions.

Three datasets that combine uniaxial and biaxial experiments are generated and contemplated in order to investigate the required and relevant amount of data. From biaxial experiments we get

$$f_{rem}(\boldsymbol{\sigma}^{bi}_{\theta *}(\sigma^{b*}_{\theta})) = -f_{mod}(\boldsymbol{\sigma}^{bi}_{\theta *}(\sigma^{b*}_{\theta}))\,, \qquad * = T,\, C, \tag{17}$$

while from their uniaxial counterparts we obtain

$$f_{rem}(\boldsymbol{\sigma}^{uni}_{\theta *}(\sigma^{u*}_{\theta})) = -f_{mod}(\boldsymbol{\sigma}^{uni}_{\theta *}(\sigma^{u*}_{\theta}))\,, \qquad * = T,\, C. \tag{18}$$

Combining N independent uniaxial and biaxial experiments, the dataset $\mathcal{D} = \{\boldsymbol{x}^i, f^i_{rem}\}^N_{i=1}$ with $\boldsymbol{x}^i = [\sigma^i_{xx}, \sigma^i_{yy}, \sigma^i_{xy}]$ is hence employed to create the surrogate model $\hat{f}_{rem}$ of the remainder term f_{rem} as detailed later in Section 3.

Remark 1. (Using Lankford coefficient data) We remark that there is potential to include Lankford coefficients in the training data set.
For the same set of uniaxial tests, known values $\bar{r}^{uT}_{\theta}$ and $\bar{r}^{uC}_{\theta}$ of Lankford coefficients can be determined with:

$$r_{rem}(\sigma^{u*}_{\theta}) = \bar{r}^{u*}_{\theta} - r_{mod}(\sigma^{u*}_{\theta})\,, \qquad * = T,\, C \tag{19}$$

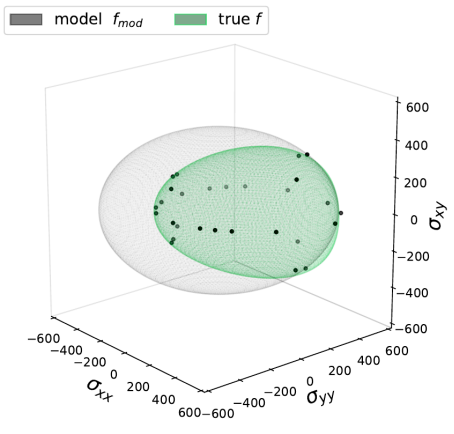
providing an implicit relationship on function derivative of $\partial f_{rem}/\partial\boldsymbol{\sigma}$ at some stress states $\boldsymbol{\sigma}^{uni}_{\theta T}$ and $\boldsymbol{\sigma}^{uni}_{\theta C}$. In order to test the capabilities of the model-data approach, we consider a simpler data-regime scenario where no experimental knowledge of the Lankford coefficients is required. This decreases the number of necessary experiments. However, available information about the Lankford coefficients could be straightforwardly included.

3 Convexity constraints and machine learning approaches

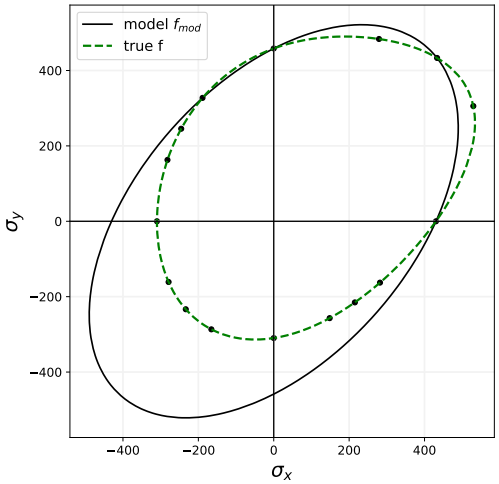
In the middle of the last century Drucker [34, 35] argued that, based on a stability postulate, the yield surface must be convex, i.e.

$$(\boldsymbol{\sigma}^{\star} - \boldsymbol{\sigma}) : \frac{\partial f}{\partial \boldsymbol{\sigma}} \leq 0 \tag{20}$$

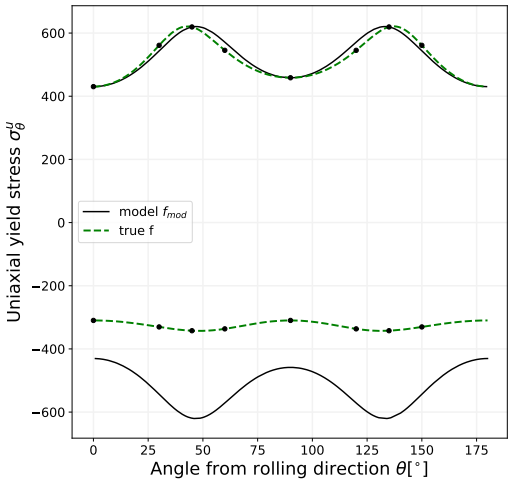
where $\boldsymbol{\sigma}^{\star}$ is an arbitrary macroscopic stress on or inside the yield surface. Hence, ideally, any yield function obtained through the model-data-driven approach of eq. (3) should fulfill this criterion. Because addition is a convexity-preserving operation, if both components are convex, the model-data-driven yield function will inherit convexity. It should be highlighted that the model component f_{mod} can directly be chosen to be, as is the case here, a convex function. Hence, the data regression model should aim to fit the best possible convex function to the available data such that convexity of the remainder term f_{rem} is guaranteed.
We remark that this is a tighter constraint than strictly necessary (it is indeed a sufficient but not necessary condition, cf. B). Different ways to enforce convexity in machine learning techniques are available, although they require some



(a) Full stress space

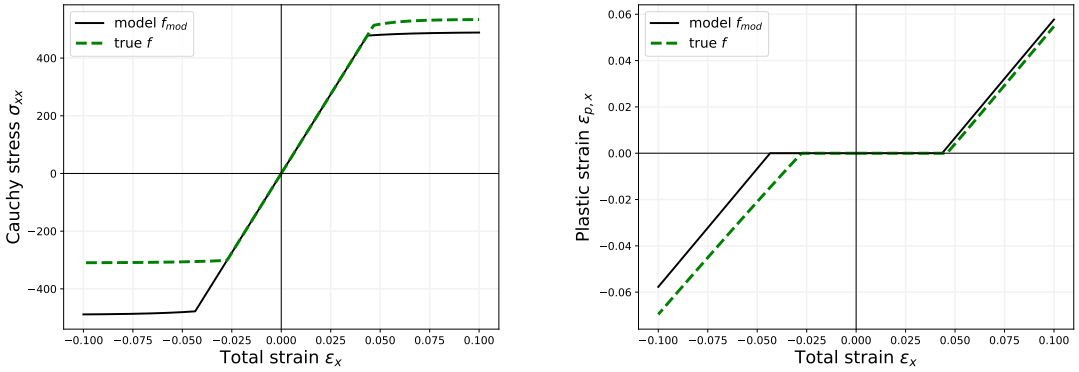


(b) Biaxial tests ($\sigma_{xy} = 0$)



(c) Uniaxial tests

Figure 1: Comparison of the yield surfaces obtained considering the model $f_{mod}(\boldsymbol{\sigma})$ (eq. (12) with parameters obtained from eq. (13)) and the true yield (obtained from the orthotropic yield criterion by Cazacu and Barlat [33], see A). Since the true yield function here follows an analytical definition, the full surface is shown, but it has to be pointed out that realistically the yield function would only be known at a few select points if it was obtained from experimental testing.



(a) (b)

Figure 2: Comparison of the responses obtained along RD only considering the model contribution $f_{mod}(\boldsymbol{\sigma})$ of the yield-function (eq. (12) with parameters obtained from eq. (13)) and the true response (obtained from the orthotropic yield criterion by Cazacu and Barlat [33], see A) during a strain-driven uniaxial traction test: (a) stress-strain relationship, (b) plastic strain.

generalizations for facing the problem at hand. Therefore, we decided to focus on the convexity condition, rather than enforcing less standard constraints on the Hessian of the surrogate model which are out of the scope of the present study.

Guaranteeing the convexity of the surrogate $\hat{f}_{rem}$ is equivalent to ensuring that the Hessian matrix of the prediction for any input $\boldsymbol{x}$ is at least positive semi-definite, i.e., the predictor is subject to the following continuous constraint

$$\boldsymbol{H}(\hat{f}_{rem})(\boldsymbol{x}) \succeq 0 \tag{21}$$

with

$$H(\hat{f}_{rem})_{ij}(\boldsymbol{x}) = \frac{\partial^2 \hat{f}_{rem}(\boldsymbol{x})}{\partial x_i \partial x_j} \tag{22}$$

where the sign " $\succeq$ " is used to symbolize positive semidefiniteness. Generally, two distinct concepts can be employed to fulfill this requirement in machine learning frameworks:

- Intrinsically convex regression functions.

  Some machine learning frameworks allow for a direct fulfillment of the convexity constraint by employing specialized designs.

- Reduction of constraints to finite dimensionality.

  Instead of ensuring the convexity requirements globally, they are only enforced (with soft or hard constraints) on a finite number of points. Different strategies have been used in the literature to choose the set of points which the requirements should be fulfilled on [36]. For instance, constraints could hold only on training samples [37, 38], on sample points in a domain of interest [39], or on a set of points that was iteratively selected [40].

The literature on convex constrained regression models using least-squares estimators is vast, see e.g. [41, 42, 43]. However, these works are mostly concerned with piecewise-linear surfaces which do not allow for the necessary differentiability at the training samples that is required for numerical evaluation of elasto-plastic calculations. In the following, convex extensions to three commonly applied smooth regression techniques are briefly discussed, in particular Support Vector Regression (SVR), Gaussian Process Regression (GPR), and Neural Networks. The first two methods are reliant on the specification of a kernel function $r : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ which is typically user chosen. The respective kernel matrix $\boldsymbol{R}(\boldsymbol{X}, \boldsymbol{Y})$ with $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{Y} \in \mathbb{R}^{m \times d}$ maps $\mathbb{R}^{n \times d} \times \mathbb{R}^{m \times d} \to \mathbb{R}^{m \times n}$ and is built with $r(\boldsymbol{X}, \boldsymbol{Y})_{i,j} = r(X_i, Y_j)$.

3.1 Input convex Support Vector Regression

In Support Vector Regression, the value of the regression model at an input $\boldsymbol{x}$ can be obtained by

$$\hat{f}_{rem}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i r(\boldsymbol{x}, \boldsymbol{x}^i) + b \tag{23}$$

where both $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]$ and b are unknown parameters. The parameters that best fit the observations are typically determined following an ϵ-insensitive loss approach [44] where no weight is put onto small residuals. This is typically done by specifying a zone $\pm\epsilon \in \mathbb{R}_+$ where the residual is not accounted for. The trainable parameters can then be found by solving a linear programming problem [45]. In order to enforce convexity of the SVR predictor in a non-intrinsic manner on a finite number of points $\{\boldsymbol{x}_c^i\}_{i=1}^{N_c}$, Wang and Ni [36] proposed to reformulate these optimization problems by including the requirement that the Hessian matrix should be positive semi-definite. Since the prediction value of eq. (23) is linear this requirement can be included such that the optimization problem takes the form of a semi-definite program

$$\begin{aligned} \underset{\boldsymbol{\alpha}, b, \boldsymbol{\xi}, \boldsymbol{a}}{\arg\min} \quad & \frac{1}{N}\mathbf{1}^T\boldsymbol{\xi} + C\mathbf{1}^T\boldsymbol{a} \\ \text{s.t.} \quad & \operatorname{diag}\left(\boldsymbol{R}\boldsymbol{\alpha} + b\mathbf{1} + \boldsymbol{\xi} - \boldsymbol{f}_{rem}, -\boldsymbol{R}\boldsymbol{\alpha} - b\mathbf{1} + \boldsymbol{\xi} + \boldsymbol{f}_{rem},\right. \\ & \left. \boldsymbol{\xi} - \mathbf{1}\epsilon, \boldsymbol{\alpha} - \boldsymbol{a}, -\boldsymbol{\alpha} + \boldsymbol{a}, \sum_{i=1}^{N} \alpha_i \boldsymbol{F}_i\right) \succeq 0 \end{aligned} \tag{24}$$

where C is used to tune the trade-off between the empirical error minimization and the regularization term maximization, and where the set of trainable parameters now includes the N-dimensional vectors $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ and

$$\boldsymbol{F}_j = \operatorname{diag}(\boldsymbol{H}(k(\boldsymbol{x}_c^1, \boldsymbol{x}_j))(\boldsymbol{x}_c^1), \ldots \boldsymbol{H}(k(\boldsymbol{x}_c^{N_c}, \boldsymbol{x}_j))(\boldsymbol{x}_c^{N_c})). \tag{25}$$

3.2 Input convex Gaussian process regression

Using some prior assumptions [46], predictions with Gaussian process regression can be made using the analytical formulation

$$\hat{f}_{rem}(\boldsymbol{x}) = \widehat{\mu} + \boldsymbol{r}_\star^T(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{f}_{rem} - \mathbf{1}\widehat{\mu}), \tag{26}$$

with $\mathbf{1}$ an all-ones vector and $\widehat{\mu} = (\mathbf{1}^T\boldsymbol{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\boldsymbol{R}^{-1}\boldsymbol{f}_{rem}$. Furthermore, the elements of $\boldsymbol{r}$ read

$$r_{\star,i}(\boldsymbol{x}) = r(\boldsymbol{x}, \boldsymbol{x}^i) \quad \text{for } i = 1, \ldots, N. \tag{27}$$

The kernel function is typically also dependent on a d-dimensional hyperparameter vector $\boldsymbol{\theta}$ which needs to be determined based on the available data, e.g. by minimizing the concentrated negative marginal log-likelihood

$$-\ln(\boldsymbol{f}_{rem}|\boldsymbol{X}, \boldsymbol{\theta})) = \frac{1}{2}\left[N\ln\left(\sigma^2(\boldsymbol{\theta})\right) + \ln(\det \boldsymbol{R}(\boldsymbol{\theta}))\right] \tag{28}$$

where

$$\widehat{\sigma}^2 = \frac{1}{N}\left(\boldsymbol{f}_{rem} - \mathbf{1}\widehat{\mu}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{f}_{rem} - \mathbf{1}\widehat{\mu}\right). \tag{29}$$

In order to constrain the prediction of eq. (26) to be strictly positive it was proposed in [39] to find the hyperparameters by postulating a constrained optimization problem on the negative marginal log-likelihood over a finite number of points $\{\boldsymbol{x}_c^i\}_{i=1}^{N_c}$. We reformulate this approach to find the hyperparameters such that the Hessian matrix of the mean prediction value is positive semi-definite on a number of constraining sample points

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\arg\min} \quad & -\ln(\boldsymbol{f}_{rem}|\boldsymbol{X}, \boldsymbol{\theta})), \\ \text{s.t.} \quad & \boldsymbol{H}(\hat{f}_{rem})(\boldsymbol{x}_c^i) \succeq 0, \qquad i = 1, \ldots, N_c \end{aligned} \tag{30}$$

A noise term can be included as a trainable parameter to improve the flexibility of the GRP regression model, see [47]. To combat that the noise of the model impacts its accuracy, [39] add an additional restriction to the constrained optimization model that penalizes noise values that would result in large discrepancies of the trained model and the actual ground truth data. This approach is also followed here. Following [39] the discrepancy parameter value is set to 0.03.

Other methods for constraining GPR exist but they generally do not guarantee that the posterior is analytically known or are only applicable for simple constraints such as output bounds. For more information and an overview over available techniques we refer to [48].

3.3 Input convex neural networks

In contrast to SVR and GPR, constraining neural networks intrinsically to be convex is possible in a simple and efficient manner to build input convex neural networks (ICNN). Following [49], the standard update formula of neural networks can be rewritten to

$$\boldsymbol{z}_{i+1} = g_i(\boldsymbol{W}_i^z \boldsymbol{z}_i + \boldsymbol{W}_i^x \boldsymbol{x} + \boldsymbol{b}_i), \tag{31}$$

where $\boldsymbol{W}_0^z = \mathbf{0}$, $\boldsymbol{z}_0 = \mathbf{0}$ and $\boldsymbol{z}_k = \boldsymbol{f}_{rem}$. The set of trainable parameters is given by $\{\boldsymbol{W}_{1:k-1}^z, \boldsymbol{W}_{0:k-1}^x, \boldsymbol{b}_{0:k-1}\}$. Contrary to traditional feed-forward neural networks, this network architecture includes "passthrough" layers, i.e. the input $\boldsymbol{x}$ is directly connected to the hidden and output layers. These layers are needed to increase the expressiveness of the network since in order to guarantee convexity with regard to the inputs all weights $\{\boldsymbol{W}_i^z\}_{i=1}^{k-1}$ are required to be non-negative and the activation functions g_i are chosen to be non-decreasing and convex. Rectified Linear Unit (ReLU) are the common choice of activation function in this context.

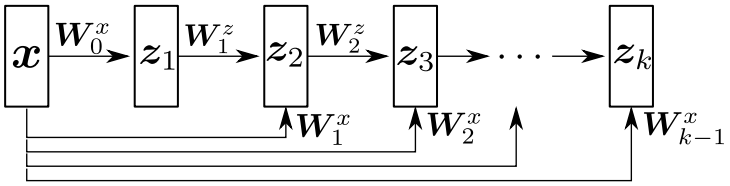


Figure 3: Visual representation of the network architecture of input convex neural networks.

The optimal values of the trainable parameters are generally determined by solving an optimization problem over a loss function where for regression problems the mean-squared error is a common choice.

4 Numerical implementation

The state $\mathcal{S}$ of an elasto-plastic material at time t is described by the current values of the total strain, as well as by a set of internal plastic variables $\mathcal{S}_{pl}(t)$, here chosen to be plastic-strain, kinematic-hardening stress variables, and scalar plastic multiplier that is

$$\mathcal{S}(t) = \{\boldsymbol{\varepsilon}(t), \mathcal{S}_{pl}(t)\} \quad \text{with} \quad \mathcal{S}_{pl}(t) = \{\boldsymbol{\varepsilon}_p(t), \lambda(t)\}\,. \tag{32}$$

For a given loading path, updating the material state requires the solution of an evolution problem, which is here faced by means of a time-incremental strategy based on the backward first-order Euler scheme. Accordingly, let us assume that all quantities are known at time t_k and denoted with subscript k, and unknowns at the next time step $t_{k+1} = t_k + \Delta t$ are indicated without subscripts.

A displacement-driven loading path is addressed. Therefore, the variation of the total strain $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_k + \Delta\boldsymbol{\varepsilon}$ is given. The evolution of material state is hence fully characterized by finding the value of the following vector of unknowns:

$$\mathbf{h} = \text{vec}\left[\{\boldsymbol{\varepsilon}_p, \lambda\}\right]\,, \tag{33}$$

which collects the values of $\mathcal{S}_{pl}$ at time t_{k+1}.

In the proposed model-data-driven strategy, updating the material state during the evolution of the plastic state, i.e., when $\mathcal{S}_{pl}(t_{k+1}) \neq \mathcal{S}_{pl}(t_k)$, requires to solve:

1. the evolution equation (4) for $\boldsymbol{\varepsilon}_p$, yielding the following algebraic expression:

$$\mathbf{Q}_\varepsilon(\boldsymbol{\varepsilon}, \mathbf{h}) = \boldsymbol{\varepsilon}_p - \boldsymbol{\varepsilon}_{p,k} - (\lambda - \lambda_k)\left[\mathbf{R}_\sigma^{mod}(\boldsymbol{\sigma}) + \mathbf{R}_\sigma^{rem}(\boldsymbol{\sigma})\right] = 0\,; \tag{34}$$

2. the yield condition:

$$q_\lambda(\boldsymbol{\varepsilon}, \mathbf{h}) = f_{mod}(\boldsymbol{\sigma}) + f_{rem}(\boldsymbol{\sigma}) = 0\,. \tag{35}$$

The dependency of $\mathbf{Q}_\varepsilon$ and q_α on $\boldsymbol{\varepsilon}$ and $\mathbf{h}$ is highlighted, since the knowledge of these latter quantities allows to uniquely characterize $\boldsymbol{\sigma}$ from Eqs. (2). The evolution of material state is then fully characterized by finding $\mathbf{h}$ such that Eqs. (34) and (35) are satisfied, that is:

$$\text{Find} \quad \mathbf{h} \quad \text{such that} \quad \mathbf{q}_{pl}(\boldsymbol{\varepsilon}, \mathbf{h}) = \text{vec}\left[\{\mathbf{Q}_\varepsilon, q_\alpha\}\right] = \mathbf{0}\,. \tag{36}$$

Algorithm 1 Pseudocode for the predictor-corrector scheme.

Trial value: $\mathbf{h}_{tr} = \mathbf{h}_k$ ▷ Initialization
if $q_\lambda(\boldsymbol{\varepsilon}, \mathbf{h}_{tr}) < 0$ **then** ▷ Predictor (elastic step)
    return with $\mathbf{h} = \mathbf{h}_{tr}$
else ▷ Corrector (plastic step)
    while $\|\mathbf{q}_{pl}(\boldsymbol{\varepsilon}, \mathbf{h}_{tr}\| > tol$ **do**
        $\Delta\mathbf{h} = -\left[\mathbb{A}(\boldsymbol{\varepsilon}, \mathbf{h}_{tr})\right]^{-1} \mathbf{q}_{pl}(\boldsymbol{\varepsilon}, \mathbf{h}_{tr})$
        $\mathbf{h}_{tr} \leftarrow \mathbf{h}_{tr} + \Delta\mathbf{h}$
    end while with $\mathbf{h} = \mathbf{h}_{tr}$
end if

The solution of the non-linear problem (36) for a given value $\boldsymbol{\varepsilon}$ is obtained by a predictor-corrector algorithm detailed in Algorithm 1. To this aim, the linearization of the algebraic expression $\mathbf{q}_{pl}(\boldsymbol{\varepsilon}, \mathbf{h})$ with respect to $\mathbf{h}$ is required, i.e.

$$\mathbb{A}(\boldsymbol{\varepsilon}, \mathbf{h}) = \frac{\partial \mathbf{q}_{pl}}{\partial \mathbf{h}} \,. \tag{37}$$

The latter can be obtained by combining partial derivatives of $\mathbf{Q}_\varepsilon$ and q_λ with respect to $\boldsymbol{\varepsilon}_p$ and λ. Most of these relationships follow standard definitions in the field of elasto-plastic numerical models, and are hence not detailed here. Only those modified by the use of the model-data-driven approach are specified here:

$$\frac{\partial \mathbf{Q}_\varepsilon}{\partial \boldsymbol{\varepsilon}_p} = \overbrace{\mathbb{I} + (\lambda - \lambda_k)\frac{\partial^2 f}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_p}}^{\mathbb{A}^{mod}_{\varepsilon\varepsilon}} + \overbrace{(\lambda - \lambda_k)\frac{\partial^2 f_{rem}}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_p}}^{\mathbb{A}^{rem}_{\varepsilon\varepsilon}} , \tag{38a}$$

$$\frac{\partial \mathbf{Q}_\varepsilon}{\partial \lambda} = \overbrace{-\frac{\partial f}{\partial \boldsymbol{\sigma}}}^{\mathbf{A}^{mod}_{\varepsilon\lambda}} \overbrace{-\frac{\partial f_{rem}}{\partial \boldsymbol{\sigma}}}^{\mathbf{A}^{rem}_{\varepsilon\lambda}} , \tag{38b}$$

$$\frac{\partial q_\lambda}{\partial \boldsymbol{\varepsilon}_p} = \overbrace{\frac{\partial f}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_p}}^{\mathbf{A}^{mod}_{\lambda\varepsilon}} + \overbrace{\frac{\partial f_{rem}}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}_p}}^{\mathbf{A}^{rem}_{\lambda\varepsilon}} , \tag{38c}$$

noting that $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}_p = -\mathbb{C}$ and $\partial q/\partial\alpha = -H$. In conclusion, the tangent operator for the plastic state evolution can be regarded as the sum of a classical one $\mathbb{A}_{mod}$ associated with the model component, and a remainder $\mathbb{A}_{rem}$ computed from the first and second derivatives of f_{rem} with respect to stress variables, cf., eq. (38). These derivatives can either be obtained as analytical derivatives of the machine learning models by hand or using automatic differentiation techniques. Since all the models used in this work were implemented in Pytorch [50] we employed automatic differentiation. For more information we refer to [51].

5 Numerical tests

The proposed approach is tested in this section. The ICNN as well as the unconstrained neural network use ReLU activation functions, are implemented in Pytorch [50] and the network parameters were optimized using the Adam optimizer [52]. The network consist of 3 hidden layers with 30 neurons each. The input convex Support Vector Regression models and their unconstrained counterparts where trained using the linear and semi-definite solvers offered by the CVXPY framework [53]. We used a fifth-order polynomial kernel for all the following results when not specified otherwise [54]. Following Wang and Ni [36] we choose $C = 10^{-4}$ and $\epsilon = 10^{-3}$.

The constrained and unconstrained Gaussian Process Regression models employ a Matérn 3/2 kernel [46]. The necessary optimization problems are solved using an interior point method [55].

All these hyperparameter and parameter choices were not the results of any elaborate study. Differences between the machine learning techniques presented in the next sections might not be accurately representing their best performances as data-driven predictive tools and should not necessarily be judged as such. Thorough studies would be required to optimize the performance of each machine learning technique, and the number of investigations and case studies to be shown would explode. Such study would hamper to reach the main take-home message of the paper, that is a proof of concept on the integration of model- and data-driven predictions. Clearly, it is however true that the

following results can be seen as a broad guidance on possible advantages and disadvantages of the presented machine learning methods by using the techniques for the same tasks and on the same datasets.

We then define three correction datasets for the yield function, c.f. eqs. (17) and (18):

$$\mathcal{D}_N = \left(\begin{bmatrix} \boldsymbol{\sigma}_{\theta*}^{uni}(\sigma_\theta^{u*}) \\ \boldsymbol{\sigma}_{\theta*}^{bi}(\sigma_\theta^{b*}) \end{bmatrix}, \begin{bmatrix} f_{rem}(\boldsymbol{\sigma}_{\theta*}^{uni}) \\ f_{rem}(\boldsymbol{\sigma}_{\theta*}^{bi}) \end{bmatrix} \right)_{\forall \theta \in \Theta_N} \tag{39}$$

consisting of $N = 28$, 16 and 8 data points respectively by defining

$$\begin{aligned} \Theta_{28} &= [0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ, 120^\circ, 135^\circ, 150^\circ] \\ \Theta_{16} &= [0^\circ, 45^\circ, 90^\circ, 120^\circ, 150^\circ] \\ \Theta_8 &= [0^\circ, 45^\circ, 90^\circ]. \end{aligned} \tag{40}$$

Hence, the subscripts on $\mathcal{D}_i$ and Θ_i refer to distinct datasets that contain 28, 16 and 8 unique data points. As for the case study in Section 2.3.1, training data is generated here by means of the orthotropic yield criterion by Cazacu and Barlat [33] (see A), so as to allow a consistent comparison of the final shape of the yield surfaces. However, this data can alternatively be straightforwardly obtained from well-established testing procedures.

In order to both decrease training time and increase training reliability [56], we employ Min-max feature scaling such that each feature lies inside the range $[0, 1]$.

In the following, $1,000$ randomly sampled points in the three-dimensional domain $[\sigma_{xx}, \sigma_{yy}, \sigma_{xy}] \in [-800, 800]^3$ MPa were used to enforce convexity in a discrete way for the surrogate modeling techniques that rely on a finite number of points where convexity should be enforced, i.e., SVR and GPR. This particular domain was chosen to strictly include all relevant information of the modeling yield function component (c.f. Figure 1(a)).

First, we study the performances of a series of machine learning techniques in a pure data-driven context for predicting the shape of the true yield function given the experimental data points. Afterwards, we compare these results to the ones obtained using the proposed machine learning techniques for the model-data-driven framework.

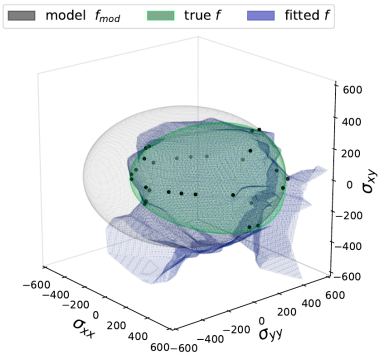
5.1 Fitting with pure data-driven approaches

In this preliminary study, we take the input data of the 28 point dataset (eq. (39)) but assume the output is the true yield function we aim to fit, i.e.

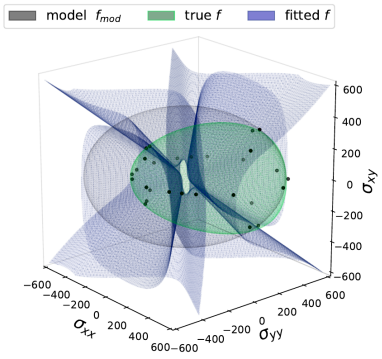
$$\mathcal{D}_{28}^\star = \left(\begin{bmatrix} \boldsymbol{\sigma}_{\theta*}^{uni}(\sigma_\theta^{u*}) \\ \boldsymbol{\sigma}_{\theta*}^{bi}(\sigma_\theta^{b*}) \end{bmatrix}, \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \right)_{\forall \theta \in \Theta_{28}} \tag{41}$$

where Θ_{28} is given in eq. (40). Among the machine learning techniques reviewed in Section 3 only Support Vector Regression and Neural Networks allow a reasonable fit of a function with a constant output (due to the hyperparameters of Gaussian Process Regression representing correlation lengths). Hence, GPR will be excluded here. Additionally, the data will be also be fitted using generalized yield function fitting via Fourier series representations. This method is summarized in C. In this method the convexity is also soft-constrained by enforcing the requirement on $1,000$ points in the relevant domain mentioned above. Here, we cut off the series after its first 6 components and solve for the trainable parameters using a stochastic gradient descent scheme (ADAM) and use $L^{0.25}$ regularization to generate sparsity in the parameter set, similarly to [57].

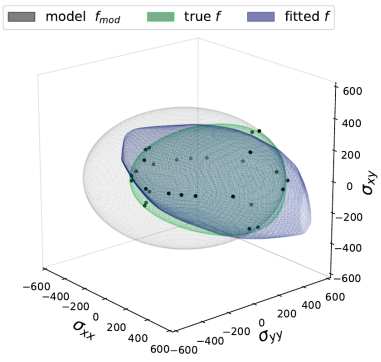
Figure 4 plots the resulting fitted yield functions employing 28 data points of $\mathcal{D}_{28}^\star$ using two unconstrained techniques (an unconstrained Neural Network and unconstrained Support Vector Regression) and three constrained approaches (the input convex Neural Networks formulation, the input convex Support Vector Regression, and the input convex Fourier series approach). It is clear that unconstrained machine learning techniques, see Figs. 4(a) and 4(b), do not generate well-defined predictive tools for yield function representations. They are accurate at predicting zero at the required input points of the dataset, however, due to the missing convexity, the scalar output value fluctuates significantly above and below zero for out-of-sample predictions, their usage appears highly compromised. The advantage of enforcing convexity is visible when looking at the remaining approaches, see Figs. 4(c) and 4(e). The yield surface shapes appear to satisfy convexity. However, we can note that none of the techniques manage to fit the true yield function in an acceptable fashion. The Fourier series representation seems to capture the anisotropy of the yield surface. Among them, the input convex Neural Network appears to generate the most accurate fit of the data points. Results obtained with input convex Neural Network will be used as benchmark performances for data-driven approaches.



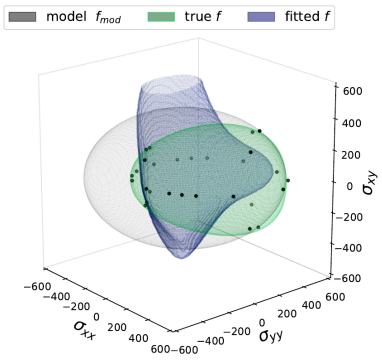
(a) Neural Network



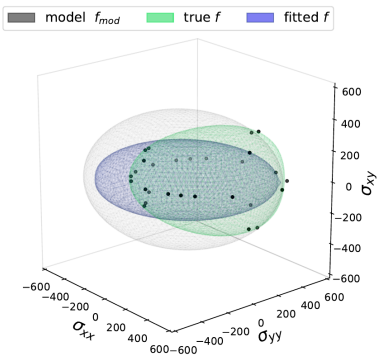
(b) Support Vector Regression



(c) Input convex neural network

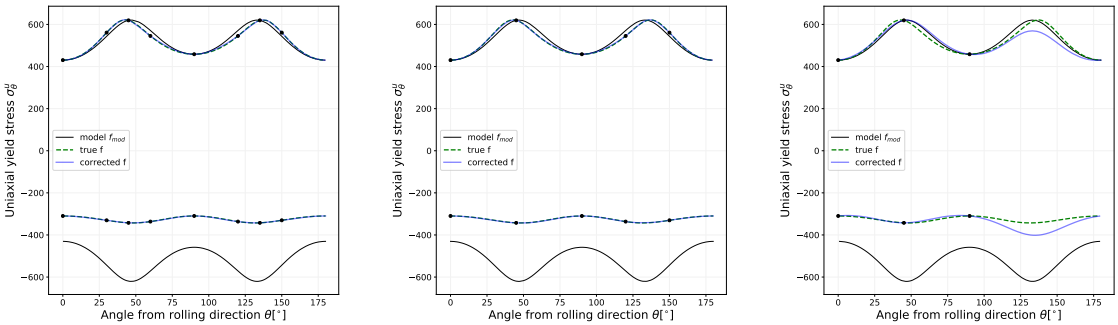


(d) Input convex Support Vector Regression

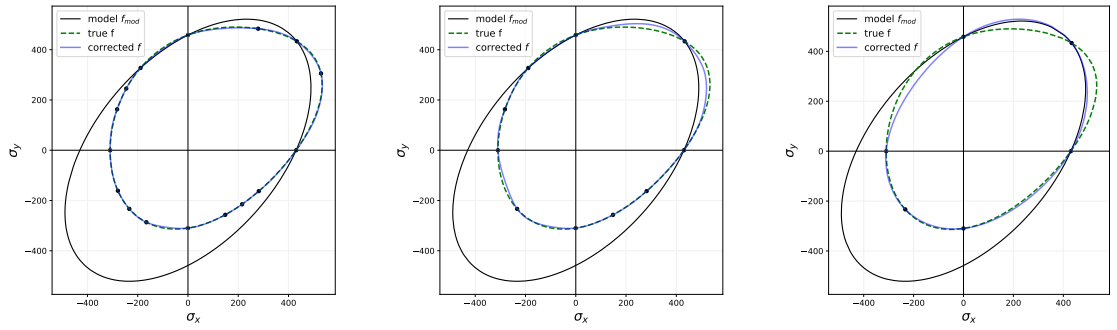


(e) Input convex Fourier series

Figure 4: Yield surface in the three-dimensional stress space using **data-driven** approaches (a-b) without convexity constraint, (c-d-e) with convexity constraint based on the experimental datasets $\mathcal{D}_{28}^{\star}$ comprising 28 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.



(a) 28 points (b) 16 points (c) 8 points



(d) 28 points (e) 16 points (f) 8 points

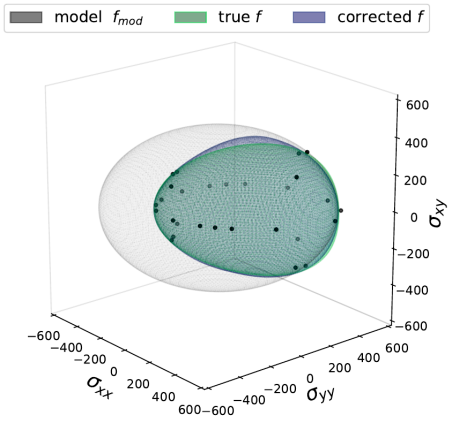
Figure 5: Yield function in uniaxial and biaxial directions using the **model-data-driven** approach with **input convex Gaussian Process Regression** as the data-driven component based on experimental datasets comprising (a,d) 28, (b,e) 16 and (c,f) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.

5.2 Fitting with model-data-driven approach

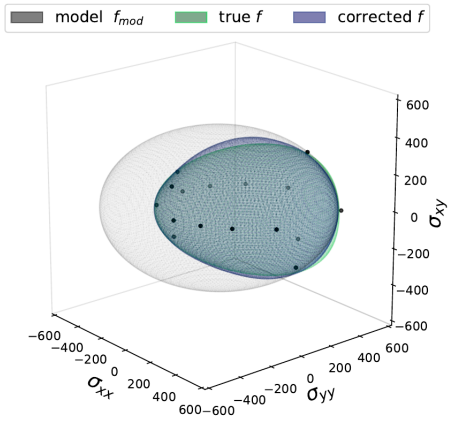
We employ the three datasets $\mathcal{D}_{28}$, $\mathcal{D}_{16}$ and $\mathcal{D}_8$ in eq. (39) to build model-data-driven yield functions based on the model component of eq. (12). Moreover, we sequentially test the performance of the three suggested machine learning approaches.

First we discuss the performance of input convex Gaussian Process Regression in the model-data-driven framework. Figure 5 plots the model-response, the response of the true model and the output of the model-data-driven approach with input convex Gaussian Process Regression in uniaxial and biaxial directions for the three datasets. The respective positions (in this projection) of the input training points are highlighted in each plot. First of all, it can be seen that by locally improving the model component the predicted output of the model-data-driven approach is able to capture the true function. Here, a larger dataset (naturally) helps to improve the accuracy of the predicted output. However, even in the case of 8 (experimental) points the difference between the model-data-driven framework and the true function is still relatively small but in regions where no data is available there are clear deviations. The yield surfaces plotted in three dimensions for Gaussian Process Regression (Figure 6) reveal that a similar pattern can be seen for the whole stress space.

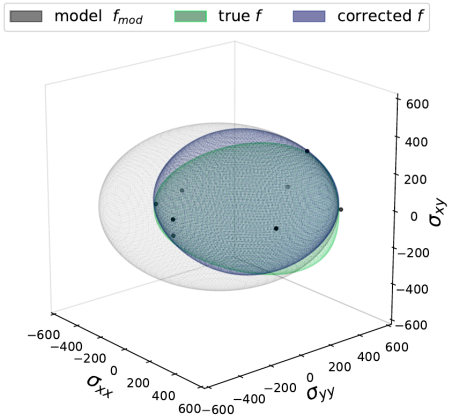
Following, the relevant uniaxial and biaxial curves using input convex Neural Networks for the correction term are compared to the true function and the modeling component in Figure 7 for all three datasets. We can highlight again that the accuracy of the modeling component is clearly improved by adding the data-driven remainder part. With a decreasing number of points the accuracy reduces which is especially noticeable in areas that are further away from the experimental data. The corresponding approximations of the yield surfaces over the full stress field using input convex Neural Networks are displayed in Figure 8.



(a) 28 points

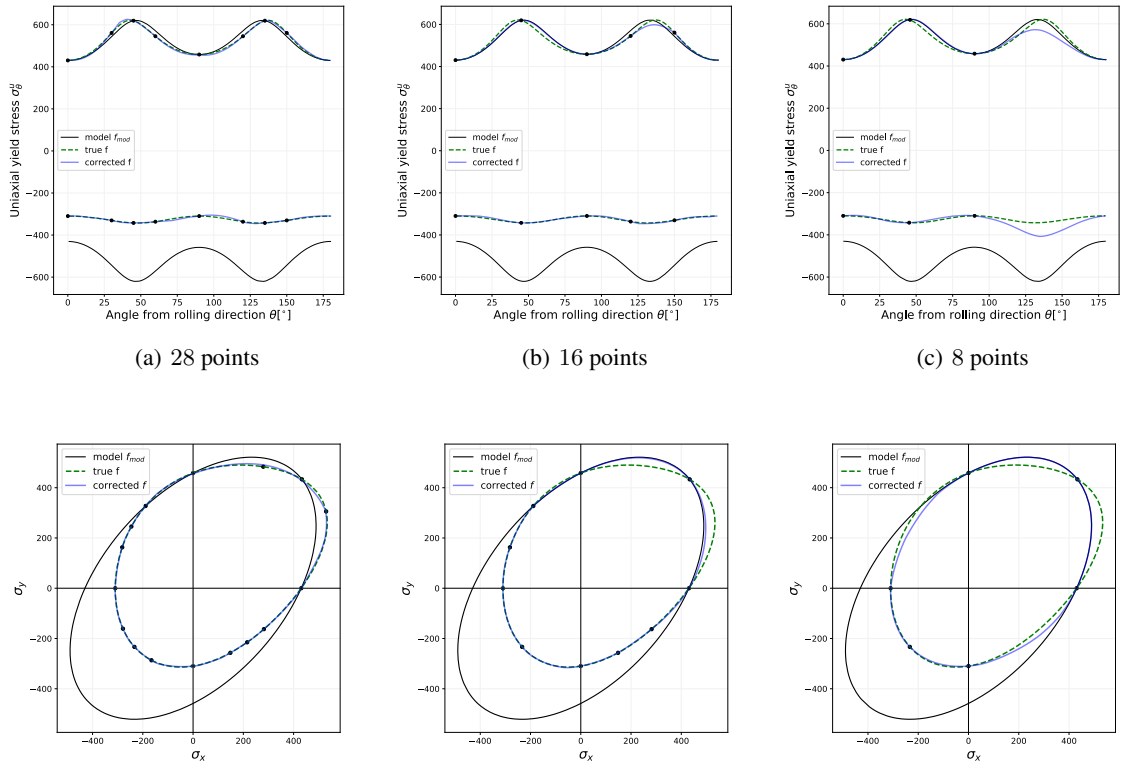


(b) 16 points



(c) 8 points

Figure 6: Yield function in the three-dimensional stress space using the **model-data-driven** approach with **input convex Gaussian Process Regression** as data-driven component based on experimental datasets comprising (a) 28, (b) 16 and (c) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.



(a) 28 points (b) 16 points (c) 8 points

(d) 28 points (e) 16 points (f) 8 points

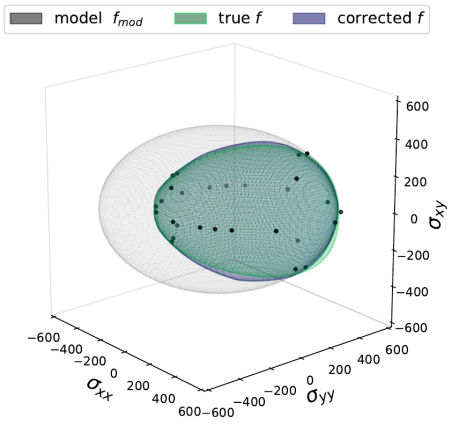
Figure 7: Yield function in uniaxial and biaxial directions using the **model-data-driven** approach with **input convex Neural Networks** as data-driven component based on experimental datasets comprising (a,d) 28, (b,e) 16 and (c,f) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.

Lastly, the same observations can be made when using input convex Support Vector Regression as the data-driven component, see Figures 9 and 10. However, we can see that, compared to the other two approaches, the accuracy of the prediction suffers more with a low amount of data.

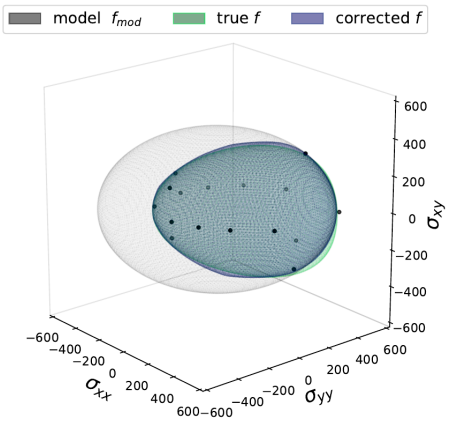
In order to highlight the accuracy of the model-data-driven approach in terms of elasto-plastic constitutive responses, Figure 11 plots the stress-strain curves for two different loading paths and for the three different input convex machine learning techniques using the $\mathcal{D}_{28}$ dataset. It can be seen that, by employing the proposed approach, the corrected curves capture more accurately the true response than the phenomenological model alone. By visual inspection the input convex Neural Network seems to perform slightly better than the other two approaches.

Consider a cycle with ϵ_{xx} oscillating from 0 to +0.1 and - 0.1, i.e. load paths as employed in Figures 11 (a-c) as reference. The average residual norm error of the applied Newton-Raphson loop over the number of iterations is plotted in Figure 12. We can see that adding a data-driven component to the phenomenological model decreases the convergence speed and final error of the root finding method. Here, input convex Neural Networks offer a performance that most closely resembles the traditional analytical model.

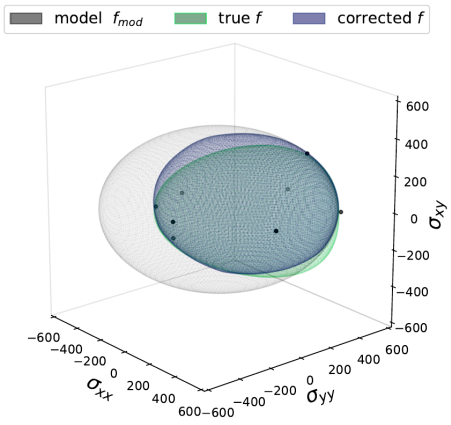
Finally, Figure 13 plots the Lankford coefficients along the rolling directions for the model, the model-data-driven approaches and the ground truth response. This allows us to investigate how well the model-data-driven framework captures the plastic anisotropy. It can be seen that, except for the low data case test of SVR, all model-data-driven yield functions improve the accuracy of this plastic flow characterization compared to the pure model component. Additionally, it is clearly visible that adding more points increases the quality of the trained yield functions.



(a) 28 points

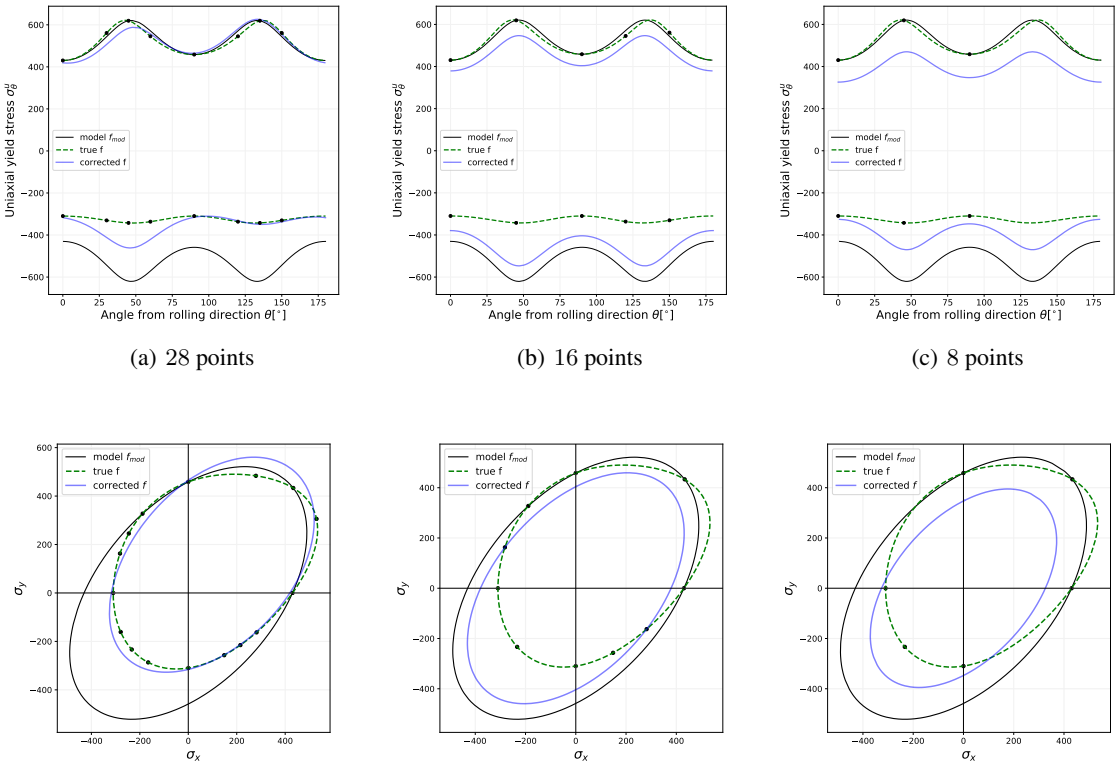


(b) 16 points



(c) 8 points

Figure 8: Yield function in the three-dimensional stress space using the **model-data-driven** approach with **input convex Neural Networks** as data-driven component based on experimental datasets comprising (a) 28, (b) 16 and (c) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.



(a) 28 points (b) 16 points (c) 8 points

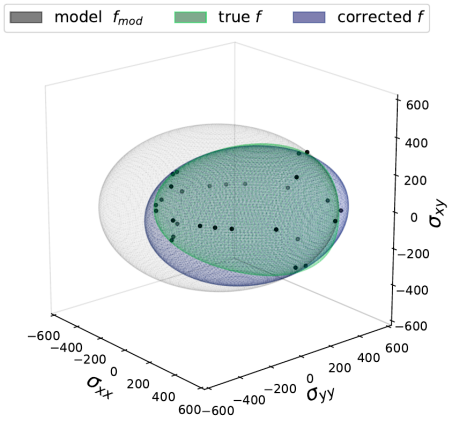
(d) 28 points (e) 16 points (f) 8 points

Figure 9: Yield function in uniaxial and biaxial directions using the **model-data-driven** approach with **input convex Support Vector Regression** as data-driven component based on experimental datasets comprising (a,d) 28, (b,e) 16 and (c,f) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.

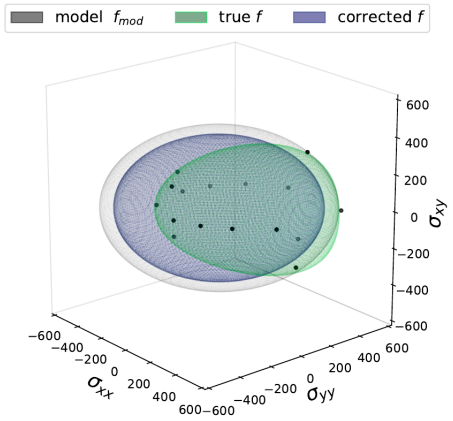
6 Discussion

The results of the previous section show that, for the chosen true response and model component assumptions, the model-data-driven component significantly improves the accuracy of the yield function compared to the pure model counterpart. Additionally, using the proposed approach, the shape of the true yield function is far better approximated in comparison with the purely data-driven techniques (cf., Fig. 4 with Figs. 6, 8 and 10). We can see that the model-data-driven prediction on 8 training points in all presented methods is significantly more accurate than the data-driven approaches fitted on 28 experimental points.

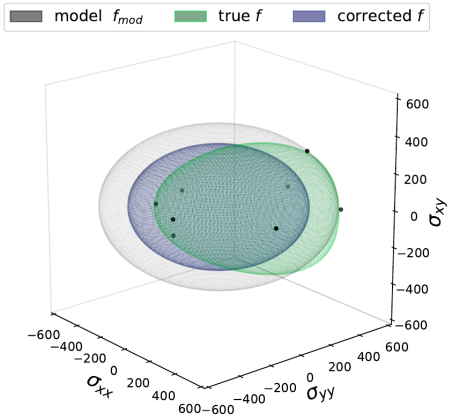
Convexity constrained Gaussian Process Regression and Neural Network seem to outperform Support Vector Regression specifically when less data is available. However, this might change when a different kernel function is applied. But based on the presented results GPR and NNs have to be the preferred choice. One major advantage of Gaussian Process Regression which has not played a role in this paper is that it allows access to posterior variances. This might be helpful in application cases where the next most effective experiments need to be located in order to improve the accuracy of the prediction based on the currently available information by using for example an active learning approach [58]. Furthermore, due to the fact that GPR is deeply rooted in statistical analysis theory, the approach enjoys access to mathematical convergence guarantees. However, as pointed out constraining a GPR model to be convex is a complex task and is only approximate. On the other hand, input convex NNs always fulfill convexity and are more expressive than GPR when a lot of data is available. Even though, using GPR for big data problems has also been explored [10]. However, the focus of the present work is experimental data which from standard measurement setups generally only allow to acquire small, restrictive data sets. Therefore, this common criticism of GPR might not be valid in the context of the present work.



(a) 28 points

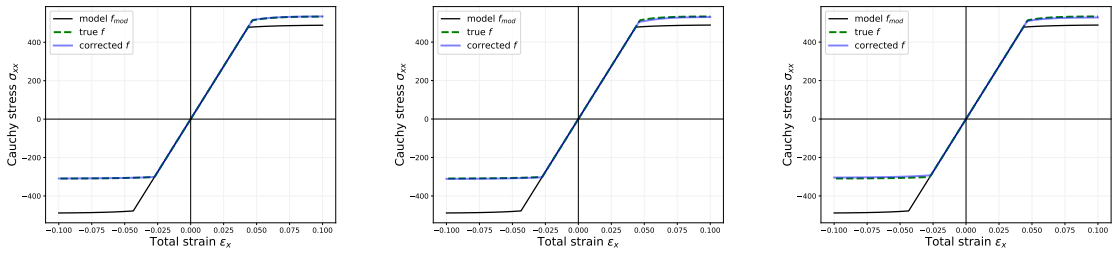


(b) 16 points

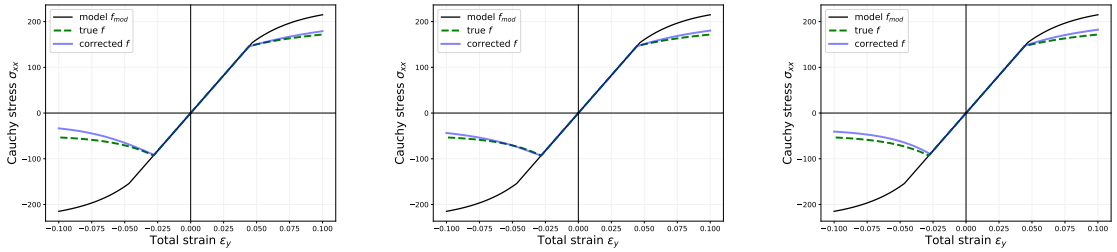


(c) 8 points

Figure 10: Yield function in the three-dimensional stress space using the **model-data-driven** approach with **input convex Support Vector Regression** as data-driven component based on experimental datasets comprising (a) 28, (b) 16 and (c) 8 points, the positions of which are visualized by dots. Comparison with true and model-driven yield functions.



(a) GPR (b) ICNN (c) SVR



(d) GPR (e) ICNN (f) SVR

Figure 11: Stress-strain curves for different machine learning techniques employing the **model-data-driven** framework with $\mathcal{D}_{28}$ dataset: (a-c) σ_{xx} over $\epsilon_{xx} \pm 0.1$, (d-f) σ_{xx} over $\epsilon_{yy} \pm 0.1$.

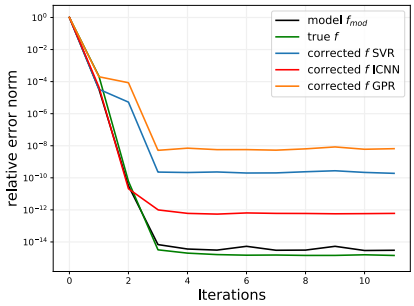
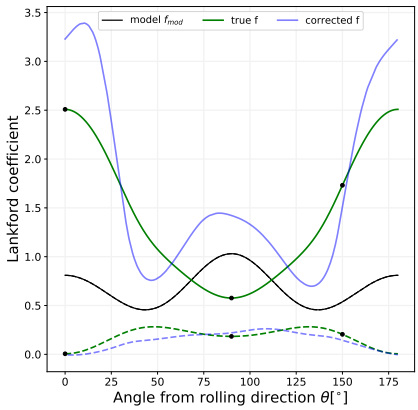
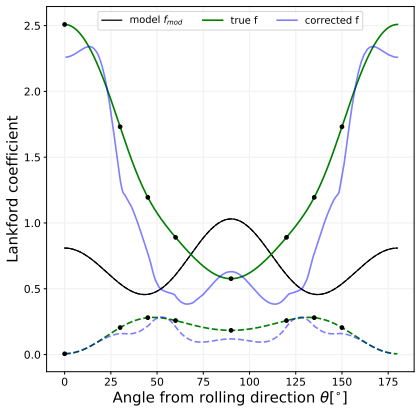


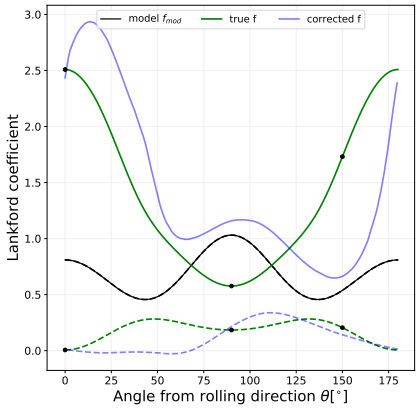
Figure 12: Average error norm values over increasing Newton-Raphson-loop iterations employing the **model-data-driven approach** with three alternative input convex machine learning techniques as data-driven component. Comparison with convergence for the true and model-driven yield functions.



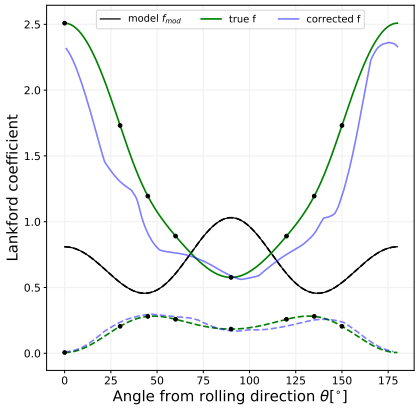
(a) GPR - 8 points



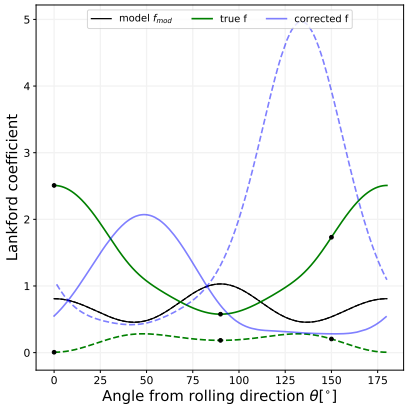
(b) GPR - 28 points



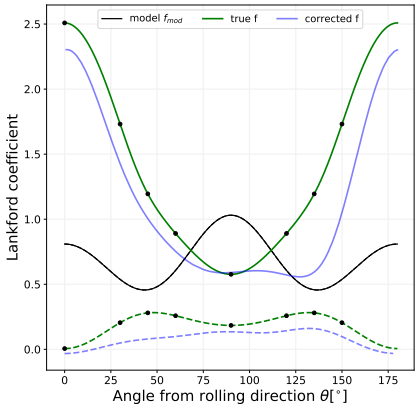
(c) ICNN - 8 points



(d) ICNN - 28 points



(e) SVR - 8 points



(f) SVR - 28 points

Figure 13: Lankford coefficient curves for different machine learning techniques employing the **model-data-driven** framework with $\mathcal{D}_8$ and $\mathcal{D}_{28}$ datasets. Comparison with the true response and the model component.

We believe that the differences between the convergence results of the Newton-Raphson loop residual stem from the different levels of fulfillment of the convexity requirement. Since, input convex Neural Networks are convex by design they show very good convergence properties when using them in gradient-based solvers. On other hand, Gaussian Process Regression and Support Vector machines are only approximately convex on a finite number of points. However, GPR has (including noise) only 4 trainable parameters whereas SVR has at least $3N + 1$ (where N is the number of data points). We speculate, that this in addition to the fact that SVR can be phrased as a simple semi-definite program which allows to exactly fulfill convexity on all constrained points allows SVR to find a function representation that is convex in a wider range of the domain than GPR, which leads to the evident difference in convergence of the residual norm error.

The predicted curves of the Lankford coefficient over the rolling direction angle (Figure 13) highlight that adding experimental values of the Lankford coefficients to the training dataset might even further increase the accuracy of the proposed approach. We will investigate this in future studies.

Overall, the framework seems to yield significant improvements compared to traditional approaches especially in the data-poor regime which is commonly encountered when physical experiments are necessary.

7 Conclusion

We present a hybrid extension to classical phenomenological model-based elasto-plasticity which allows to locally improve the chosen analytical yield function model where experimental data is available. Our framework is developed with data availability, and the idea of data-poor modeling in mind. The application case studied in this work is related to rolled metal sheets, which exhibit a clear anisotropy in the yield surface which in turn makes fitting accurate yield surfaces more complex. In order to ensure convexity of the predicted yield function in the presented approach, we establish a direct connection to the convexity of the data-driven correction component. Input convex extensions of three classical machine learning techniques are reviewed and applied to the problem at hand. It can be seen that the presented approach significantly improves the accuracy of the standalone model component while achieving better performances than purely data-driven models as well. Furthermore, we are able to generate surprisingly accurate yield surfaces in the data-poor regime.

In future works, the framework can be applied to other use cases which might include a dependence of the yield function on hydrostatic pressure and on actual experimental data. Furthermore, we envision using adaptive sampling techniques in order to determine the best possible next experiment [58] that allows to improve the representation of the yield function the most. Additionally, we aim to extend this framework by adding hardening.

Overall, we believe that the presented approach shows a lot of potential to tackle problems in elasto-plasticity modeling where data availability is a concern.

Acknowledgments

M. Marino gratefully acknowledge funding from the Italian Ministry of Education, University and Research MIUR (Programma per Giovani Ricercatori - anno 2017 Rita Levi Montalcini) and of Regione Lazio (POR FESR LAZIO 2014; Progetti di Gruppi di Ricerca 2020; project: BIOPMEAT, n. A0375-2020-36756). N. Bouklas and J.N. Fuhg gratefully acknowledge support by the Air Force Office of Scientific Research under award number FA9550-22-1-0075.

References

[1] Amir Barati Farimani, Joseph Gomes, and Vijay S Pande. Deep learning the physics of transport phenomena. *arXiv preprint arXiv:1709.02432*, 2017.

[2] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1633–1642, 2019.

[3] Teeratorn Kadeethum, Daniel O'Malley, Jan Niklas Fuhg, Youngsoo Choi, Jonghyun Lee, Hari S Viswanathan, and Nikolaos Bouklas. A framework for data-driven solution and parameter estimation of pdes using conditional generative adversarial networks. *Nature Computational Science*, 1(12):819–829, 2021.

[4] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[5] Jan N Fuhg and Nikolaos Bouklas. The mixed deep energy method for resolving concentration features in finite strain hyperelasticity. *Journal of Computational Physics*, 451:110839, 2022.

[6] Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, 2019.

[7] Jan Niklas Fuhg, Ioannis Kalogeris, Amélie Fau, and Nikolaos Bouklas. Interval and fuzzy physics-informed neural networks for uncertain fields. *Probabilistic Engineering Mechanics*, page 103240, 2022.

[8] Mauricio Fernández, Mostafa Jamshidian, Thomas Böhlke, Kristian Kersting, and Oliver Weeger. Anisotropic hyperelastic constitutive models for finite deformations combining material theory and data-driven approaches with application to cubic lattice metamaterials. *Computational Mechanics*, 67(2):653–677, 2021.

[9] Jan Niklas Fuhg and Nikolaos Bouklas. On physics-informed data-driven isotropic and anisotropic constitutive models through probabilistic machine learning and space-filling sampling. *arXiv preprint arXiv:2109.11028*, 2021.

[10] Jan N Fuhg, Michele Marino, and Nikolaos Bouklas. Local approximate gaussian process regression for data-driven constitutive models: development and comparison with neural networks. *Computer Methods in Applied Mechanics and Engineering*, 388:114217, 2022.

[11] David González, Francisco Chinesta, and Elías Cueto. Learning corrections for hyperelastic models from data. *Frontiers in Materials*, 6:14, 2019.

[12] Jan Niklas Fuhg, Christoph Böhm, Nikolaos Bouklas, Amelie Fau, Peter Wriggers, and Michele Marino. Model-data-driven constitutive responses: application to a multiscale computational framework. *International Journal of Engineering Science*, 167:103522, 2021.

[13] Ari Frankel, Craig M Hamel, Dan Bolintineanu, Kevin Long, and Sharlotte Kramer. Machine learning constitutive models of elastomeric foams. *Computer Methods in Applied Mechanics and Engineering*, 391:114492, 2022.

[14] W Wu and D Kolymbas. Hypoplasticity then and now. In *Constitutive modelling of granular materials*, pages 57–105. Springer, 2000.

[15] Rodney Hill. *The mathematical theory of plasticity*, volume 11. Oxford university press, 1998.

[16] J Ghaboussi, JH Garrett Jr, and Xiping Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics*, 117(1):132–153, 1991.

[17] J Ghaboussi and DE Sidarta. New nested adaptive neural networks (nann) for constitutive modeling. *Computers and Geotechnics*, 22(1):29–52, 1998.

[18] M Mozaffar, R Bostanabad, W Chen, K Ehmann, Jian Cao, and MA Bessa. Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences*, 116(52):26414–26420, 2019.

[19] Dengpeng Huang, Jan Niklas Fuhg, Christian Weißenfels, and Peter Wriggers. A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 365:113008, 2020.

[20] Nikolaos N Vlassis and WaiChing Sun. Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering*, 377:113695, 2021.

[21] Jan N Fuhg, Lloyd van Wees, Mark Obstalecki, Paul Shade, Nikolaos Bouklas, and Matthew Kasemer. Machine-learning convex and texture-dependent macroscopic yield from crystal plasticity simulations. *arXiv preprint arXiv:2202.01885*, 2022.

[22] Rubén Ibáñez, Emmanuelle Abisset-Chavanne, David González, Jean-Louis Duval, Elias Cueto, and Francisco Chinesta. Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. *International Journal of Material Forming*, 12(4):717–725, 2019.

[23] Alexander Hartmaier. Data-oriented constitutive modeling of plasticity in metals. *Materials*, 13(7):1600, 2020.

[24] Hyungbum Park and Maenghyo Cho. Multiscale constitutive model using data–driven yield function. *Composites Part B: Engineering*, 216:108831, 2021.

[25] GF Bomarito, TS Townsend, KM Stewart, KV Esham, JM Emery, and JD Hochhalter. Development of interpretable, data-driven plasticity models with symbolic regression. *Computers & Structures*, 252:106557, 2021.

[26] Moritz Flaschel, Siddhant Kumar, and Laura De Lorenzis. Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering*, 381:113852, 2021.

[27] J. Neggers, O. Allix, F. Hild, and S. Roux. Big Data in Experimental Mechanics and Model Order Reduction: Today's Challenges and Tomorrow's Opportunities. *Archives of Comptational Methods in Engineering*, 25(1):143–164, 2018.

[28] Dorel Banabic, Frédéric Barlat, Oana Cazacu, and Toshihiko Kuwabara. Advances in anisotropy of plastic behaviour and formability of sheet metals. *International Journal of Material Forming*, 13(5):749–787, 2020.

[29] Siyuan Fang, Xiaowan Zheng, Gang Zheng, Boyang Zhang, Bicheng Guo, and Lianxiang Yang. A new and direct r-value measurement method of sheet metal based on multi-camera dic system. *Metals*, 11(9):1401, 2021.

[30] Oana Cazacu, Benoit Revil-Baudard, and Nitin Chandola. *Plasticity-damage couplings: from single crystal to polycrystalline materials*. Solid Mechanics and Its Applications. Springer, 2019.

[31] Oana Cazacu and Frédéric Barlat. Generalization of drucker's yield criterion to orthotropy. *Mathematics and Mechanics of Solids*, 6(6):613–630, 2001.

[32] Rodney Hill and Egon Orowan. A theory of the yielding and plastic flow of anisotropic metals. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 193(1033):281–297, 1948.

[33] Oana Cazacu and Frederic Barlat. A criterion for description of anisotropy and yield differential effects in pressure-insensitive metals. *International Journal of Plasticity*, 20(11):2027–2045, 2004. Daniel C. Drucker Memorial Issue.

[34] Daniel Charles Drucker. A more fundamental approach to plastic stress-strain relations. In *Proc. of 1st US National Congress of Applied Mechanics, 1951*, pages 487–491, 1951.

[35] Daniel Charles Drucker. A definition of stable inelastic material. *Journal of Applied Mechanics*, 26(1):101–106, 1959.

[36] Yongqiao Wang and He Ni. Multivariate convex support vector regression with semidefinite programming. *Knowledge-Based Systems*, 30:87–94, 2012.

[37] Kristiaan Pelckmans, Marcelo Espinoza, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Primal-dual monotone kernel regression. *Neural Processing Letters*, 22(2):171–182, 2005.

[38] Pang Du, Christopher F Parmeter, and Jeffrey S Racine. Nonparametric kernel regression with multiple predictors and multiple shape constraints. *Statistica Sinica*, pages 1347–1371, 2013.

[39] Andrew Pensoneault, Xiu Yang, and Xueyu Zhu. Nonnegativity-enforced gaussian process regression. *Theoretical and Applied Mechanics Letters*, 10(3):182–187, 2020.

[40] Klaus Neumann, Matthias Rolf, and Jochen Jakob Steil. Reliable integration of continuous constraints into extreme learning machines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21(supp02):35–50, 2013.

[41] Timo Kuosmanen. Representation theorem for convex nonparametric least squares. *The Econometrics Journal*, 11(2):308–325, 2008.

[42] Emilio Seijo and Bodhisattva Sen. Nonparametric least squares estimation of a multivariate convex regression function. *The Annals of Statistics*, 39(3):1633–1657, 2011.

[43] Eunji Lim and Peter W Glynn. Consistency of multidimensional convex regression. *Operations Research*, 60(1):196–208, 2012.

[44] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[45] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[46] Jan N Fuhg. Adaptive surrogate models for parametric studies. *arXiv preprint arXiv:1905.05345*, 2019.

[47] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[48] Laura P Swiler, Mamikon Gulian, Ari L Frankel, Cosmin Safta, and John D Jakeman. A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 2020.

[49] Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[51] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[53] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[54] Shahaboddin Shamshirband, Dalibor Petković, Hossein Javidnia, and Abdullah Gani. Sensor data fusion by support vector regression methodology—a comparative study. *IEEE Sensors Journal*, 15(2):850–854, 2014.

[55] Arkadi S Nemirovski and Michael J Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[56] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[57] Moritz Flaschel, Siddhant Kumar, and Laura De Lorenzis. Discovering plasticity models without stress data. *arXiv preprint arXiv:2202.04916*, 2022.

[58] Jan N Fuhg, Amélie Fau, and Udo Nackenhorst. State-of-the-art and comparative review of adaptive sampling methods for kriging. *Archives of Computational Methods in Engineering*, 28(4):2689–2747, 2021.

[59] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[60] Stefan Cristian Soare and AA Benzerga. On the modeling of asymmetric yield functions. *International Journal of Solids and Structures*, 80:486–500, 2016.

A Orthotropic yield criterion by Cazacu and Barlat [33]

Cazacu and Barlat [33] propose an anisotropic and asymmetric yield criterion of the form

$$f_{true} = (J_2^0)^{3/2} - cJ_3^0 - \bar{\sigma}^3\,, \tag{42}$$

where $c \in \mathbb{R}$, J_2^0 is defined in eq. (11) and

$$\begin{aligned} J_3^0 = &\frac{1}{27}(b_1+b_2)\sigma_{xx}^3 + \frac{1}{27}(b_3+b_4)\sigma_{yy}^3 + \frac{1}{27}\left[2(b_1+b_4)-b_2-b_3\right]\sigma_{zz}^3 \\ &+ 2b_{11}\sigma_{xy}\sigma_{xz}\sigma_{yz} + \frac{2}{9}(b_1+b_2)\sigma_{xx}\sigma_{yy}\sigma_{zz} - \frac{1}{9}(b_1\sigma_{yy}+b_2\sigma_{zz})\sigma_{xx}^2 \\ &- \frac{1}{9}(b_3\sigma_{zz}+b_2\sigma_{xx})\sigma_{yy}^2 - \frac{1}{9}\left[(b_1-b_2+b_4)\sigma_{xx} + (b_1+b_3+b_4)\sigma_{yy}\right]\sigma_{zz}^2 \\ &- \frac{\sigma_{yz}^2}{3}\left[(b_6+b_7)\sigma_{xx} - b_6\sigma_{yy} - b_7\sigma_{zz}\right] - \frac{\sigma_{xz}^2}{3}\left[2b_9\sigma_{yy} - b_8\sigma_{zz} - (2b_9-b_8)\sigma_{xx}\right] \\ &- \frac{\sigma_{xy}^2}{3}\left[2b_{10}\sigma_{zz} - b_5\sigma_{yy} - (2b_{10}-b_5)\sigma_{xx}\right] \end{aligned} \tag{43}$$

where $\{a_i\}_{i=1}^6$ (in J_2^0) and $\{b_i\}_{i=1}^{11}$ are two sets of anisotropic coefficients that can be determined when data of experimental yield loci is available.
For thin-sheet metals these sets reduce to $\{a_i\}_{i=1}^3$ and $\{b_i\}_{i=1}^4$. The authors get them and the additional unknown value c from experimental yield loci data for Mg-4% Li alloys (see Table 1). The yield function estimated by eq. (A) is assumed as the ground truth model in the current study.

a_1	a_2	a_3	b_1	b_2	b_3	b_4	c	$\bar{\sigma}$
0.896	3.371	3.509	-1.591	5.414	3.957	0.259	2.01	300 MPa

Table 1: Anisotropy coefficients and values for c and $\bar{\sigma}$ for Mg-4% Li alloy as established in [33].

B Convexity requirements on yield function

Consider a split of the yield function into the model and the remainder part

$$f(\boldsymbol{\sigma}) = f_{mod}(\boldsymbol{\sigma}) + f_{rem}(\boldsymbol{\sigma}). \tag{44}$$

Assume a two-dimensional stress space and define $\boldsymbol{x} = [\sigma_x, \sigma_{xy}, \sigma_y]$. The yield function is convex if its symmetric Hessian matrix $\boldsymbol{H}$ with

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{45}$$

is at least positive semi-definite which can be denoted by

$$\boldsymbol{H}(f) \succeq 0. \tag{46}$$

This is equivalent to (see for example [59])

$$\boldsymbol{a}^T \boldsymbol{H}(f)\boldsymbol{a} \geq 0, \qquad \forall \boldsymbol{a} \in \mathbb{R}^3. \tag{47}$$

Since the model component is user-chosen, we can pick a convex function f_{mod}. Hence, we know that

$$\boldsymbol{a}^T \boldsymbol{H}(f_{mod})\boldsymbol{a} \geq 0, \qquad \forall \boldsymbol{a} \in \mathbb{R}^3. \tag{48}$$

In order to ensure the convexity of f, the remainder term has then to fulfill the following condition

$$\begin{aligned} \boldsymbol{a}^T \boldsymbol{H}(f_{mod})\boldsymbol{a} + \boldsymbol{a}^T \boldsymbol{H}(f_{rem})\boldsymbol{a} &\geq 0 \\ \boldsymbol{a}^T \boldsymbol{H}(f_{rem})\boldsymbol{a} &\geq -\boldsymbol{a}^T \boldsymbol{H}(f_{mod})\boldsymbol{a} \end{aligned} \tag{49}$$

for all $\boldsymbol{a} \in \mathbb{R}^3$. To (computationally) simplify this constraint, we tighten the condition on the remainder term to the following instead

$$\begin{aligned} &\boldsymbol{a}^T \boldsymbol{H}(f_{rem})\boldsymbol{a} \geq 0 \\ \iff &\boldsymbol{H}(f_{rem}) \succeq 0 \end{aligned} \tag{50}$$

which includes the condition of eq. (49).

C Fitting of generalized yield function with Fourier series

Recently, there have been attempts in the literature to fit yield surface data based on Fourier series representations [60, 57]. The regression model is of the following form

$$f = \sqrt{\frac{3}{2}} r - \sum_{i=0}^{M} \left[\alpha_i \cos(i\alpha) + \beta_i \sin(i\alpha)\right] \tag{51}$$

with

$$r = \sqrt{\pi_1^2 + \pi_2^2}, \qquad \alpha = \arctan 2(\pi_2, \pi_1) \tag{52}$$

and

$$\pi_1 = \sqrt{\frac{2}{3}}\sigma_1 - \sqrt{\frac{1}{6}}\sigma_2 - \sqrt{\frac{1}{6}}\sigma_3, \qquad \pi_2 = \sqrt{\frac{1}{2}}\sigma_2 - \sqrt{\frac{1}{2}}\sigma_3. \tag{53}$$

Here $\sigma_1 \leq \sigma_2 \leq \sigma_3$ are the principal stresses in increasing order and $\{\alpha_i, \beta_i\}_{i=1}^{M}$ with $M \geq 0$ are the trainable parameters. Here M is a user-chosen positive integer value that sets the number of used terms in the approximation.