Computers and Structures 252 (2021) 106557

Contents lists available at ScienceDirect

Computers and Structures

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

Development of interpretable, data-driven plasticity models with symbolic regression

G.F. Bomarito [a,*], T.S. Townsend [b], K.M. Stewart [c], K.V. Esham [d], J.M. Emery [e], J.D. Hochhalter [c]

[a] NASA Langley Research Center, Hampton, VA, USA
[b] University of Central Florida, Orlando, FL, USA
[c] University of Utah, Salt Lake City, UT, USA
[d] The Ohio State University, Columbus, OH, USA
[e] Sandia National Laboratories, Albuquerque, NM, USA

ARTICLE INFO

Article history:
Received 18 February 2021
Accepted 7 April 2021

Keywords:
Plasticity
Homogenization
Symbolic regression
Genetic programming

ABSTRACT

In many applications, such as those which drive new material discovery, constitutive models are sought that have three characteristics: (1) the ability to be derived in automatic fashion with (2) high accuracy and (3) an interpretable nature. Traditionally developed models are usually interpretable but sacrifice development time and accuracy. Purely data-driven approaches are usually fast and accurate but lack interpretability. In the current work, a framework for the rapid development of interpretable, data-driven constitutive models is pursued. The approach is characterized by the use of symbolic regression on data generated with micromechanical finite element models. Symbolic regression is the search for equations of arbitrary functional form which match a given dataset. Specifically, an implicit symbolic regression technique is developed to identify a plastic yield potential from homogenized finite element response data. Through three controlled test cases of varying complexity, the approach is shown to successfully produce interpretable plasticity models. The controlled test cases are used to investigate the robustness and scalability of the method and provide reasonable recommendations for more complex applications. Finally, the recommendations are used in the application of the method to produce a porous plasticity model from data corresponding to a representative volume element of voids within a metal matrix.

Published by Elsevier Ltd.

1. Introduction

Though occasionally being criticized as less axiomatic than balance laws (e.g., mass, momentum), the development of constitutive models has helped enable the use of new materials and alloys. Specifically, elastoplastic constitutive models based on Tresca [1], Mises [2], and Hill [3] yield functions have stood as the bedrock for engineering analyses over the years. These three models, specifically focusing on plastic deformation, have reached this status because they posses two qualities: (1) they are accurate and tunable enough to be predictive, and (2) they are interpretable, allowing for good judgment in application and extension.

In contrast to the above constitutive models, which are rooted in physics and extended to fit observations, there has been a trend toward data-driven constitutive model development. Generally speaking these models embrace the notion that accuracy and predictive capability are more important then having a sound understanding of the physics involved. Neural network based constitutive models [4–6] are one example of this paradigm, wherein a neural network model is trained to predict a response (e.g., stress) as a function of load (e.g., deformation). The resulting neural net is often referred to as a *black-box* owing to its lack of physical insightfullness and interpretability. Another extreme of the data-driven paradigm is the recent work in *data-driven computational mechanics* which removes the idea of constitutive models altogether and use data directly for the constitutive relationship [7–9]. This method, by definition, is a *black-box* and offers no more insight into the constitutive relation than the underlying data itself.

There have also been examples of data-driven constitutive models that fall somewhere in the middle of the interpretablity spectrum. For instance, the work of Soare et al. [10], Soare and Barlat [11] that uses relatively opaque, higher-order polynomials as a plasticity model; however, the derivation and fitting of the coefficients preserves a well known aspect of plastic yield functions: convexity. Another example of partially interpretable constitutive models are based on hybrid modeling [e.g. 12], wherein the

[*] Corresponding author.
E-mail address: geoffrey.f.bomarito@nasa.gov (G.F. Bomarito).

https://doi.org/10.1016/j.compstruc.2021.106557
0045-7949/Published by Elsevier Ltd.

modeling error of a classical physics-based model is corrected using a *black-box* model.

Physics-informed machine learning is another paradigm that combines interpretable aspects of physics with *black-box* modeling [13–15]. In this variant of machine learning, physics are incorporated into the objective function for training of a *black-box* model; either in the form of a residual of a partial differential equation [14] or the weak form of the equivalent boundary value problem [15]. Because the approach in the current work also uses physics in training, it can be viewed as a physics-informed machine learning method; however, there are two key differences between the current work and other physics-informed machine learning works. Firstly, in contrast to hidden physics models [13] where coefficients in a fixed equation are found, a model is sought with arbitrary functional form. And secondly, the models that are produced in the current work are interpretable (in the form of relatively simple equations).

In many – if not most – cases the use of *black-box* models causes no issues, so long as the model is predictive. There are, however, advantages to interpretable plasticity models. Most importantly, interpretable constitutive models (1) can help drive knowledge discovery (2) may be used more confidently outside the range of their training data and (3) are usually easier to implement within current computational mechanics frameworks.

A common tool for the construction of interpretable, data-driven models is symbolic regression (SR). SR is a procedure for rapidly finding equations that fit a given dataset. Owing to recent advances in scalability, robustness, and efficiency [16–20], SR has reached maturity such that it can be used in many complex, real-wold applications. For example, Schmidt and Lipson [21] developed a method for performing implicit SR to discover conservation laws from experimental data. The focus of the current paper is the application of SR to the development of interpretable plasticity models. The two primary contributions of the current paper are (1) the formulation of the plasticity modeling problem as an implicit symbolic regression problem and (2) a framework for obtaining the relevant training data from micromechanical simulation.

The topic of using SR in the development of constitutive models has been investigated before. One early example is the work of Schoenauer et al. [22], which developed a three-dimensional hyperelastic model in terms of a strain energy function. The model had limited success for two reasons. First, inserting physical constraints upon the possible strain energy function proved difficult. Second, the presence of real-valued constants within the generated strain energy expressions caused difficulty in the SR optimization. The presence of real-valued constants have been an obstacle for the general SR community [23,24]. Since the time of Schoenauer et al. [22], techniques have been developed to address the issue, such as restriction of the constants to coefficients of linearly combined SR terms (which allows for regression to obtain best-fit constants) [25,26] as well as embedded nonlinear optimization of constants [27]. The latter is more computationally intensive but does not restrict the functional form obtained with SR.

For the most part, SR has been used in plasticity model development as a tool for modifying sub-components of a traditionally developed model. Examples include parametrically homogenized constitutive models, which use SR as a means of tying tunable parameters in a yield function to microstructural features [28–30]. Another example is the work of Versino et al. [31] who used SR in the creation of a flow stress model as part of a Mises (J_2) plasticity framework. They worked with relatively sparse experimentally-based datasets, because designing and performing experiments that could capture the needed data was costly and difficult. They were able to overcome the difficulty by supplementing their datasets with synthetic data points which represented their expert knowledge. Nonetheless, generation of significant quantities of relevant experimental data can be a challenge. For some applications there are high data-output experimental techniques [32–34] to feed data-hungry SR; however, in the current work an alternative approach is taken wherein data for SR is derived from micromechanical finite element (FE) simulations. The primary benefit is that simulation data is usually easier to generate than experimental data. Because the data is more plentiful, more complex relationships can be investigated. The underlying assumption of such an approach is that the micromechanical model accurately represents the material of interest.

In the current work, a framework is developed for using SR in the formulation of plastic yield functions based on the homogenized responses of micromechanical FE models. Details involved in the construction of the micromechanical FE models, along with the synthesis of their response data with SR, are covered in Section 2. The approach is then tested in three test cases (Sections 3 and 4) illustrating its ability to generate interpretable and accurate plasticity models. Section 5 then discusses some of the key aspects that affect the accuracy and robustness of the method. Using the insights produced in Section 5, the method is applied to the problem of porous plasticity in Section 6. The major findings of this work are summarized in Section 7.

2. Approach

The approach taken in the current work is a type of automatic multi-scale computational homogenization. Multi-scale computational homogenization is characterized by the coarse-graining of the response of a representative volume element (RVE) to a larger scale [e.g., 35–37]. It requires that the RVE be well-characterized and that the scale of the RVE is significantly separated from the scale where the homogenized constitutive model is used (see the work of Geers et al. [38] for more details).

The two main components of the approach are (1) data collection using FE simulation and (2) plasticity model synthesis with SR. The plasticity model developed by SR is henceforth referred to as the homogenized model.

2.1. Data collection using FE simulation

The goal of this step is to generate data associated with the plastic response of a material by using a micromechanical model. The micromechanical model consists of an FE model of an RVE of the material in question. The RVE is subjected to a set of n_L mechanical loading cases $L^{(i)}$ for $i \in \{1, 2, \ldots, n_L\}$, where each loading case consists of n_s load steps. The type of loading is chosen to be quasi-static mechanical loading of the RVE because a rate-independent mechanical plasticity model is the ultimate goal of the current work; however, other physical phenomena could be investigated by adjusting the types of loading encountered (e.g., thermal loading). For example, loading cases which aim to explore rate-dependent plasticity could be conducted at several loading rates. In general terms, the loading applied to the RVE should be representative of the types of loadings for which the homogenized model is intended to be used.

In the current work, loads are applied to the RVE through displacement boundary conditions on an initially cuboid RVE. The displacement loading conditions are chosen to achieve a prescribed change in the homogenized deformation gradient $\Delta\bar{\boldsymbol{F}}$ between each load step; thus, a loading case is characterized by n_s values of $\Delta\bar{\boldsymbol{F}}$: i.e.,

$$L^{(i)} \equiv \{\Delta\bar{\boldsymbol{F}}^{(i1)}, \Delta\bar{\boldsymbol{F}}^{(i2)}, \ldots, \Delta\bar{\boldsymbol{F}}^{(in_s)}\} \tag{1}$$

or more compactly

$$L^{(i)} \equiv \{\Delta\bar{\boldsymbol{F}}^{(ij)}\}, \tag{2}$$

with $j \in \{1, 2, \ldots, n_s\}$. Note that overbar notation is used to signify homogenized quantities of the RVE, and that bold face is used to refer to second order tensors.

At every step in every loading case, several quantities are post-processed from the simulated RVE results. Homogenized stress $\bar{\boldsymbol{\sigma}}$ is calculated as the volume average of the stress within the RVE. The homogenized strain $\bar{\epsilon}$ is calculated based on the motion of the boundaries, or equivalently using the deformation gradient obtained by integrating $\Delta\bar{\boldsymbol{F}}$ through to the given load step. Lastly, any measures of state that can be quantified from the RVE should be included; inclusion of measurable internal variables, $\bar{\theta}$, allows for the homogenized SR model to be dependent upon those variables. The included variables will vary greatly depending on the RVE, but for mechanical plasticity models this should at least include some measure of plastic/dissipated work. In an RVE for a porous plasticity model [e.g. 35,36], void volume fraction and void shape would be examples of other measurable internal variables that could be included from the RVE.

Collecting the above measured quantities from the RVE results in a dataset that can be used in SR to characterize the constitutive response of the RVE. The dataset consists of:

$$\bar{\boldsymbol{\sigma}}^{(ij)}, \quad \bar{\boldsymbol{\epsilon}}^{(ij)}, \quad \bar{\theta}_1^{(ij)}, \quad \bar{\theta}_2^{(ij)}, \quad \ldots \quad \bar{\theta}_{n_\theta}^{(ij)} \tag{3}$$

for $i = 1, \ldots, n_L$ and $j = 1, .., n_s$. n_θ is the number of measurable internal variables included from the RVE.

2.2. Plasticity model synthesis with symbolic regression

SR is the process of searching for a function $f(X) \in \mathbb{R}$ that best describes a given dataset. In order to perform the search, fitness must be defined with respect to the dataset. After selecting a fitness metric, a strategy for searching the space of real-valued equations must be chosen.

2.2.1. Fitness metric definition

There are several strategies and fitness measures that could be used for the formulation of plasticity models. The method pursued in the current work is based on SR of a yield function which is dependent upon stress $\boldsymbol{\sigma}$ and n_θ internal variables θ, i.e., $\Phi(\boldsymbol{\sigma}, \theta_1, \ldots \theta_{n_\theta})$. The yield function is defined as:

$$\Phi(\boldsymbol{\sigma}, \theta_1, \ldots \theta_{n_\theta}) = 0. \tag{4}$$

Assuming that the dataset used in this step has a measure of plastic work, the dataset can be filtered to include data from only the steps which involved plastic deformation. By further assuming that all of the remaining steps satisfy Eq. (4) (i.e., all steps where plastic deformation occur end on the yield surface), an implicit SR problem is defined. Implicit SR will be covered in more depth below. It is assumed that the above yield equation can be found with SR and the elastic behavior of the material is known. The last component that must be considered is the evolution of the internal variables. n_θ variable evolution equations must also be found with SR; they are of the form:

$$\dot{\theta}_m = \hat{f}(\dot{\boldsymbol{\epsilon}}, \dot{\theta}_1, \ldots \dot{\theta}_{m-1}), \tag{5}$$

with $1 \leqslant m \leqslant n_\theta$. Note that each internal variable evolution can be a function of all preceding internal variables. Development in this way ensures that all internal variable evolution can be calculated without simultaneous solution of multiple (likely nonlinear) equations. Of course, effort must be made to optimize the ordering of the variables, and iteration may be required. Eq. (5) can be reorganized as

$$\dot{\theta}_m - \hat{f}(\dot{\boldsymbol{\epsilon}}, \dot{\theta}_1, \ldots \dot{\theta}_{m-1}) = 0 \tag{6}$$

$$f(\dot{\boldsymbol{\epsilon}}, \dot{\theta}_1, \ldots \dot{\theta}_m) = 0, \tag{7}$$

in which case another set of implicit SR problems are defined.

The above formulation of the fitness metrics is intentionally general, and needs only minor modifications to be applied to other types of loading. The key here is that $\boldsymbol{\sigma}$ fully captures the loading (i.e. rate-independent) that is being explored in the current work. Extensions to other loading types would include the necessary loading descriptors in place of (or in addition to) $\boldsymbol{\sigma}$.

Implicit SR is the search for invariant equations within a dataset; it is defined by the general form:

$$f(X) = constant, \tag{8}$$

where X is the set of parameters in the input dataset: $X = \{x_0, x_1, \ldots\}$. Examples of effective solution of implicit SR have been seen in the literature [21,39,40]. The crux of the solution of implicit SR is the definition of fitness such that trivial solutions, such as $f(X) = 0.0(x_0)$, are avoided. Schmidt and Lipson [21,39] used implicit derivatives to achieve this, and Hills et al. [40] used the principle of least action in addition to an explicit normalization term. In the current work, a new fitness metric for implicit regression was formulated drawing inspiration from the above examples and the Prager consistency condition. The fitness metric was developed by first taking the time derivative of Eq. (8):

$$\frac{df}{dt} = 0. \tag{9}$$

Expanding the derivatives leads to

$$\frac{df}{dx_k}\frac{dx_k}{dt} = 0, \tag{10}$$

where the summation across the index is intended. The $\frac{df}{dx_k}$ terms can be calculated via automatic differentiation from any candidate equation. As in the works of Schmidt and Lipson [21,39], the input data will be assumed to take the form of several trajectories with respect to time $x_k(t)$. Time derivatives of each trajectory $\frac{dx_k}{dt}$ can be estimated through nonparametric fitting [41]; the current work uses 3rd order polynomials and a window size of 7 time steps. The sum of absolute values of the time derivative terms is used for normalization, which helps avoid trivial solutions to the implicit regression problem, as follows

$$\frac{\frac{df}{dx_k}\frac{dx_k}{dt}}{|\frac{df}{dx_\ell}||\frac{dx_\ell}{dt}|} = 0. \tag{11}$$

Because the above equation holds for all times and loading trajectories, the fitness metric E is defined as

$$E(f, X) = \frac{1}{n_L n_s}\sum_{i=1}^{n_L}\sum_{j=1}^{n_s}\left|\frac{\frac{df}{dx_k}\frac{dx_k}{dt}}{|\frac{df}{dx_\ell}||\frac{dx_\ell}{dt}|}\right|^{(ij)}. \tag{12}$$

This can be described in general terms as the mean normalized rate of deviation of f along all the loading trajectories in the dataset. The practical implementation of this fitness function requires the consideration of non-finite values such as NaN (not a number) and inf (infinity), that may appear as a result of the normalization or function evaluation. Non-finite values are filtered from the mean calculation: i.e., the non-finite (ij) terms are dropped and the leading divisor updated accordingly. If less than 90% of the (ij) terms are finite, the fitness function is evaluated as inf.

Anecdotally, the above fitness metric was seen to have both lower computation time and more effective convergence on the implicit regression problems of the current work than the implicit derivatives scheme of Schmidt and Lipson [21,39]. Further work needs to be performed to show in which cases this is true, such as higher problem dimensionality, etc.

2.2.2. Equation search with genetic programming

There are several methods for searching the space of real-valued equations [e.g., 42,25,43]; genetic programming (GP) is among the most popular. GP involves the application of genetic algorithms to search a space of computer programs. The use of GP in the current work is very similar to the one outlined by Schmidt and Lipson [21] with the inclusion of a number of other more recent advances. The key portions of the GP search algorithm are covered below in more detail.

An acyclic graph representation was chosen for the candidate equations. The acyclic graph form has been shown to have comparable convergence characteristics to the more common tree representation but with better performance characteristics [44]. An example of an acyclic graph representation of the equation $f(X) = x_1 - (3.14x_0 + \sin x_0)$ is illustrated in Fig. 1(a). The acyclic graph is implemented as a sequence of mathematical operations (see Fig. 1(b)) where the result of each operation is stored in a buffer that can be later referenced by other operations. The result of the last operation in the list is that function's evaluation. It is important to note that not every operation in the operation list will contribute to the function evaluation (i.e., some portions of the acyclic graph may be disconnected from the main tree). This is why a portion of the 16 element operation list in Fig. 1(b) is excluded. Though these operations may not contribute to the evaluation of that function, they can act as a carrier of genetic information for future generations. The complexity of an equation is defined as the number of operations that contribute to the function's evaluation: e.g., the function in Fig. 1(a) has a complexity of 7. Graphs that represent the same function with lower complexity are described as more succinct.

Mutation and crossover are mechanisms by which GP evolves candidate solutions through modification of the operation list. Single point crossover, where the beginning portion of one operation list is combined with the end portion of another, occurs with a prescribed probability p_x. A mutation probability p_m is also prescribed. If mutation occurs, one of three types of mutation are used: (1) mutation of an operator and its parameters, (2) mutation of just the parameters of an operation, and (3) pruning/removal of a branch. The choice is made between the three types of mutations by random choice with probabilities p_{m_1}, p_{m_2}, and p_{m_3}, respectively.

Numerical constants within the function are represented abstractly as $c_0, c_1, \ldots$. Similar to the work of Kommenda et al. [27], the constants within a function are optimized such that the fitness function is minimized for the given functional form. Every time the function is modified to a form which affects the number of constants, the optimization procedure reoccurs. The Levenberg–Marquardt [45] algorithm is used for the optimization process; it has been shown to be one of the most effective algorithms for constant optimization within the GP framework [46]. A random initial guess between the bounds of $[-100, 100]$ is used as a starting point for the optimization.

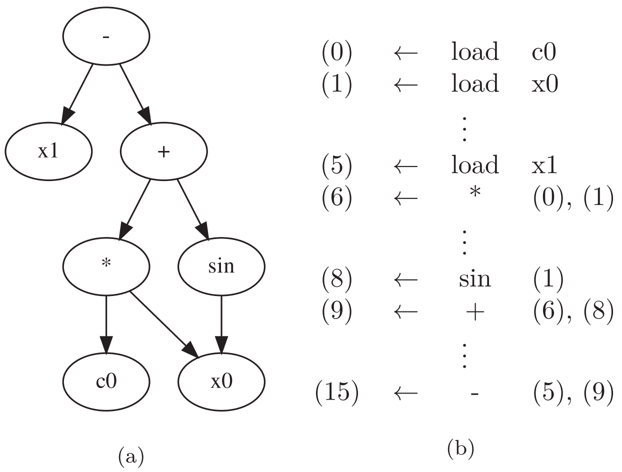


Fig. 1. Representation of the equation $f(X) = x_1 - (c_0x_0 + \sin x_0)$ as an acyclic graph viewed as both (a) a connected graph and (b) an operation list. Note that the operations which do not contribute to the function's evaluation are excluded from the operation list.

Because GP is computationally expensive, several other efforts are made here to improve performance and robustness of the GP algorithm, namely coevolution of fitness predictors [18], age-fitness Pareto optimization [19], and parallel island evolution [47]. A coevolution of fitness predictors scheme is used to improve efficiency. As in the work of Schmidt and Lipson [18], a subset predictor is used wherein fitness is evaluated based on only a portion of the input data, which effectively acts as a surrogate for the true fitness (the fitness that would be calculated using all of the data). A budget of approximately 10% of the total computational effort is devoted to the evolution of the fitness predictors. Age-fitness Pareto optimization is incorporated to minimize the risk of premature convergence. Deterministic crowding [48] was also tested with similar effectiveness in large population cases. An island population division [e.g., 49,47] was used in order to allow for the use of distributed computing systems. Each sub-population (henceforth referred to as an island) evolves independently on separate processors. Migration (random shuffling of populations) between random pairs of islands occurs at set intervals of n_M generations.

The several parameters which define the GP algorithm are included in Table 1. Except where otherwise stated, these parameters are used for all GP runs in the current work. The values of these hyperparameters were chosen based on recommendations and examples in the literature [18,19,47] which illustrated their effective use in discovery of equations with complexities similar to those sought in the current work. And since the specific choice of hyperparameters is expected to influence performance of SR more than its overall ability to produce reasonably fitting equations, a comprehensive hyperparameter study is considered beyond the scope of the current work. It is worth noting that hyperparamer optimization schemes (e.g. random/grid search) that are often applied to other machine learning methods can also be applied to SR. Due to longer run times, however, it may be worth tuning the hyperparameters of SR on a representative problem with faster evaluation time.

3. Test cases

The above approach is illustrated in three verification test cases. In all test cases, a single-element FE model with an isotropic elastic–plastic material model is used as a material point evaluator

Table 1
Genetic programming algorithm parameters.

Parameter	Value
Number of islands	40–160
Mathematical operators	$+ - \times$
Solution (equation) population per island	128
Fitness predictor population per island	32
Fitness predictor trainer population per island	40
Acyclic graph's operation list length	128
Fitness predictor subset size	128
Crossover probability, p_x	0.7
Mutation probability, p_m	0.01
Mutation type probabilities, $p_{m_1}, p_{m_2}, p_{m_3}$	0.4, 0.4, 0.2
Migration rate in number of generations, n_M	1000

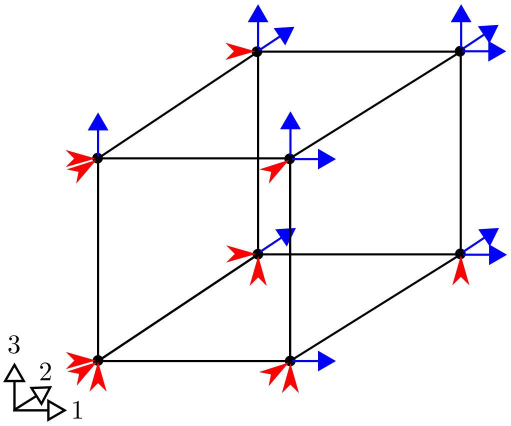


Fig. 2. The single 8-node finite element MPE used in the test cases with illustrated boundary conditions. Fixed degrees of freedom are indicated by a red arrow in the direction of the fixity. Blue arrows indicate applied displacements. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(MPE, see Fig. 2). A known plasticity model is prescribed to the element and then the model is subjected to a mechanical load. The effectiveness of the above approach is gauged by its ability to recover the prescribed plasticity model from the load-response data of the MPE.

The elastic portion of the model is described by a Poisson's ratio of 0.3 and an elastic modulus of 1000[1]. The von Mises plasticity model is chosen for the plastic portion of the material model in all test cases. It is described by the plastic potential:

$$\Phi = (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 - 2\sigma_y^2 = 0. \tag{13}$$

Here, $\sigma_1, \sigma_2, \sigma_3$ are the three principle stresses and σ_y is the current yield stress. Since the von Mises model is isotropic, only three values are needed to completely characterize stress or strain. Thus, homogenized values of stress $\bar{\boldsymbol{\sigma}}$ and strain $\bar{\boldsymbol{\epsilon}}$ from the MPE are then fully characterized by their principle components $\bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3$ and $\bar{\epsilon}_1, \bar{\epsilon}_2, \bar{\epsilon}_3$. The principle components of the homogenized stress are used as inputs into the SR algorithm with the target of finding an equation of the form of Eq. (13). Note that other characterizations of the stress could be used, such as the three stress invariants I_1, I_2, and I_3, in which case a different but equivalent form of Eq. (13) would be the target.

The single internal variable associated with the above von Mises plasticity model is equivalent plastic strain ϵ_{vm}. The equivalent plastic strain is used to help describe the evolution of the yield stress, i.e., hardening behavior. Thus, the current yield stress is a function of the equivalent plastic strain: $\sigma_y(\epsilon_{vm})$. Note that some other measure of plastic work would function equally well for this purpose.

For all test cases $n_L = 70$ and $n_s = 33$; these values were chosen based on SR efficiency in a parametric study (discussed Section 5.2). Two-stage loading is utilized where $j \leqslant \frac{n_s}{5}$ corresponds to stage 1 and $j > \frac{n_s}{5}$ corresponds to stage 2. Each stage consists of proportional loading from its starting point; thus, $\Delta\bar{\boldsymbol{F}}^{(ij)}$ is constant during each loading stage. A random orientation of the loading is chosen for stage 1 of each loading case and stage 2 is defined by a permutation of stage 1:

$$\Delta\bar{\boldsymbol{F}}^{(ij)} = \begin{cases} \frac{F_{max}}{n_s} \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & f_3 \end{bmatrix} : j \leqslant \frac{n_s}{5} \\ \frac{F_{max}}{n_s} \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_3 & 0 \\ 0 & 0 & f_1 \end{bmatrix} : j > \frac{n_s}{5} \end{cases} \tag{14}$$

with f_1, f_2, and f_3 drawn from a uniform random distribution $(0, 1)$. Note again that the choice of distributions of the loading should be representative of the loading cases for which the model is intended to be used. For instance, the $(0, 1)$ bounds on the uniform distribution limit the loadings to be primarily tensile, whereas bounds of $(-1, 1)$ would allow for tensile and comprehensive loadings. In the same way, a different choice the distributions of f_i would adjust the concentration the loadings.

The choice of stage 2 loading as a permutation of stage 1 is arbitrary; a loading in which stage 2 is chosen from three additional random variables is speculated to perform equally well. The motivation for using two-stage loading is, firstly, to illustrate how SR can handle complex loading cases. Secondly, proportional loading with von Mises plasticity can result in parallel plastic load paths, as illustrated in Fig. 3. Including only parallel segments in the implicit SR described in Section 2.2.1 gives rise to an infinite set of trivial solutions. The trivial solutions would be any potential which is constant in the direction of the segments; two examples of such potentials are illustrated in Fig. 4. Using non-proportional loading avoids this issue[2]. In general, non-proportional loading is not a requirement of this method but rather addressing a specific quirk of the von Mises model. However, more complex loading types are expected to more fully exercise an RVEs response compared to proportional loading.

Each of the three test cases describes an SR setup for the development of a plastic yield function. Each test case is uniquely defined by its input dataset and its target outcome.

3.1. Test case 1: von Mises yield equation without hardening

In test case 1, an elastic-perfectly plastic model is used, which means that the current yield stress is constant and equal to the initial yield stress: $\sigma_y = \sigma_{y0}$. In the absence of hardening, the von Mises yield equation describes a cylinder in principle stress space, with its axis colinear with the hydrostatic $\sigma_1 = \sigma_2 = \sigma_3$ axis (illustrated in Fig. 5). For test case 1, the yield stress is $\sigma_{y0} = 10$.

The input dataset for test case 1 consists of the homogenized principle stresses of the single element MPE $\bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3$ in addition to its equivalent plastic strain, ϵ_{vm}. In formal terms, the input dataset for test case 1 is described as:

$$X_1 = \left\{ \bar{\sigma}_1^{(ij')}, \bar{\sigma}_2^{(ij')}, \bar{\sigma}_3^{(ij')}, \epsilon_{vm}^{(ij')} \right\}. \tag{15}$$

Note that, as described in Section 2.2.1, the dataset is filtered so as to include only loading cases that include active plasticity: i.e., $ij' = \{i, j \in [1, n_L], [1, n_s] : \epsilon_{vm}^{(ij)} > \epsilon_{vm}^{(ij-1)}\}$. An exemplary subset of the dataset is illustrated in Fig. 5.

Substituting the homogenized principle stresses of the MPE into Eq. (13) and noting that $\sigma_y = constant$ gives the target for SR in test case 1:

$$(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_3 - \bar{\sigma}_1)^2 = constant. \tag{16}$$

The objectives of test case 1 are to establish the ability of SR to:

[1] Units for all example problems are arbitrary, but in practice should remain self-consistent.

[2] There are alternative methods to combat trivial soutions caused by parallel load paths. For example, a term could be added to the fitness that would penalize the residual of Eq. (4) or the variance of Eq. (8); however, these solutions have the undesireable effect of encouraging the magnitude of f to be extremely small.

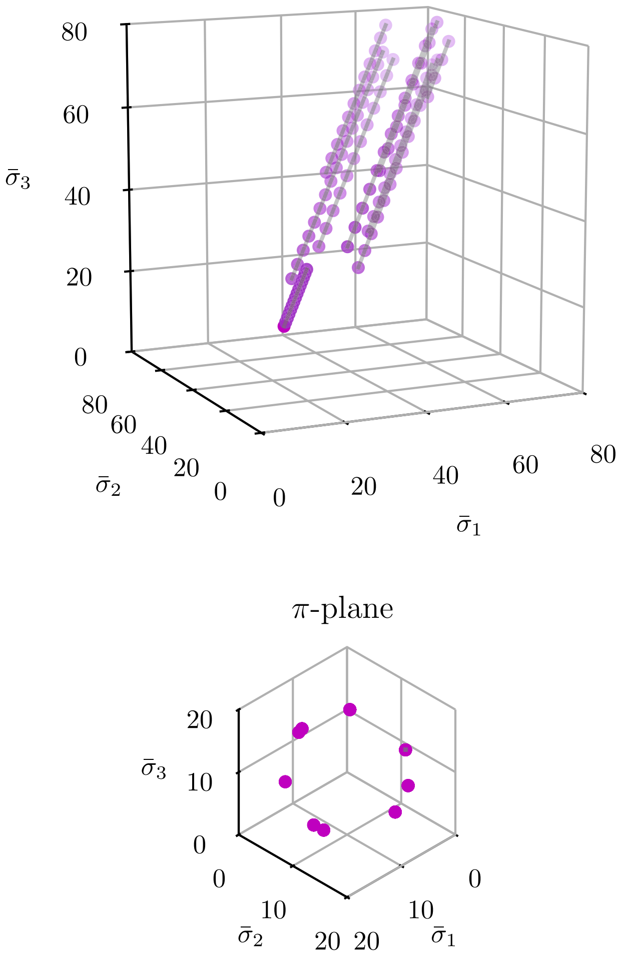


Fig. 3. Example of MPE response data, depicting how proportional loadings can result in parallel load paths. The load paths are all parallel to each other and are perpendicular to the π-plane. Load steps are represented as points. Loading paths are represented as lines connecting the points.

- Discover a basic plastic yield function from MPE response data.
- Discern which variables from the input dataset are useful in the yield function and which are not, i.e., ϵ_{vm} is not present in the above equation even though it is provided to the SR algorithm.

3.2. Test case 2: von Mises yield equation with hardening

In test case 2, an elastic–plastic model with linear hardening is used; thus, the current yield stress is described by a linear relation to the equivalent plastic strain:

$$\sigma_y = \sigma_{y0} + H\epsilon_{vm}. \tag{17}$$

For test case 2, the initial yield stress is $\sigma_{y0} = 10$ and the hardening parameter is $H = 990$.

The input dataset for test case 2 consists of the same elements as test case 1; however, the quantities are gathered from an MPE with linear hardening:

$$X_2 = \left\{ \bar{\sigma}_1^{(ij')}, \bar{\sigma}_2^{(ij')}, \bar{\sigma}_3^{(ij')}, \epsilon_{vm}^{(ij')} \right\}. \tag{18}$$

A subset of the dataset is illustrated in Fig. 6. Instead of a cylinder, each segment in the dataset now traverses a portion of a cone.

Again, substituting the homogenized values from the MPE into Eq. (13) gives:

$$(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_3 - \bar{\sigma}_1)^2 - 2\sigma_y^2 = 0. \tag{19}$$

Substituting in Eq. (17) and simplifying further gives the target equation for test case 2:

$$\begin{aligned}&(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_3 - \bar{\sigma}_1)^2 - c_1\epsilon_{vm} - c_2\epsilon_{vm}^2 \\ &\quad = constant,\end{aligned} \tag{20}$$

where $c_1 = 4\sigma_{y0}H$ and $c_2 = 2H^2$. Substituting the provided values of σ_{y0} and H give c_1 and c_2 values of $39{,}600^2$ and $1{,}960{,}200^2$, respectively.

The objectives of test case 2 are to establish the ability of SR to:

- Discover a more complex plastic yield function from MPE response data.
- Identify a functional form that requires the use of real-valued numerical constants. This task has been historically difficult for SR.

3.3. Test case 3: equivalent plastic strain evolution equation

An important difference between test cases 1 and 2 is the presence of the equivalent plastic strain (an internal variable) as part of the yield function in test case 2. The plasticity model in test case 2 is incomplete without an appropriate evolution equation for that variable. The MPE for test case 3 uses the same linear hardening material model from test case 2. But, test case 3 focuses on finding the evolution equation for ϵ_{vm}.

Because ϵ_{vm} is the only internal variable under consideration, the implicit SR problem defined by Eq. (7) requires only the strain rates of the MPE and the equivalent plastic strain rate. It is assumed that the elastic behavior of the MPE is known. The elastic portion of the strain can thus be removed, giving the plastic strain, $\bar{\boldsymbol{\epsilon}}_p$, and corresponding plastic strain rate, $\dot{\bar{\boldsymbol{\epsilon}}}_p$. Incremental forms of these variables are used in the test case 3 dataset:

$$X_3 = \left\{ \Delta\bar{\epsilon}_{p1}^{(ij')}, \Delta\bar{\epsilon}_{p2}^{(ij')}, \Delta\bar{\epsilon}_{p3}^{(ij')}, \Delta\bar{\epsilon}_{vm}^{(ij')} \right\}, \tag{21}$$

where $\bar{\epsilon}_{p1}, \bar{\epsilon}_{p2}$, and $\bar{\epsilon}_{p3}$ are the three principle components of the plastic strain. A backwards difference is used to define each incremental quantity, i.e., $\Delta\bar{\epsilon}_{p1}^{(ij)} = \bar{\epsilon}_{p1}^{(ij)} - \bar{\epsilon}_{p1}^{(ij-1)}$.

The evolution equation for equivalent plastic strain in von Mises plasticity is written as:

$$\dot{\epsilon}_{vm} = \sqrt{\frac{2}{9}\left[(\dot{\epsilon}_{p1} - \dot{\epsilon}_{p2})^2 + (\dot{\epsilon}_{p2} - \dot{\epsilon}_{p3})^2 + (\dot{\epsilon}_{p3} - \dot{\epsilon}_{p1})^2\right]}. \tag{22}$$

Inserting incremental volume-averaged values and reorganizing leads to the target equation for test case 3:

$$\begin{aligned}&(\Delta\bar{\epsilon}_{p1} - \Delta\bar{\epsilon}_{p2})^2 + (\Delta\bar{\epsilon}_{p2} - \Delta\bar{\epsilon}_{p3})^2 \\ &+(\Delta\bar{\epsilon}_{p3} - \Delta\bar{\epsilon}_{p1})^2 - \tfrac{9}{2}\Delta\bar{\epsilon}_{vm}^2 = constant.\end{aligned} \tag{23}$$

The objectives of test case 3 are to establish the ability of SR to:

- Discover the internal variable evolution equation which is complementary to test case 2. These two components would then fully define a homogenized plasticity model derived by SR.
- Find another functional form that requires the use of numerical constants.
- Find an accurate functional form given data with some amount of noise. The noise in this case comes from inaccuracies in the derivative calculations with backwards difference.

4. Test case results

All SR test cases were repeated 20 times to ensure robustness and repeatability. In each SR run the solution populations are evolved until convergence. Convergence is defined as coming

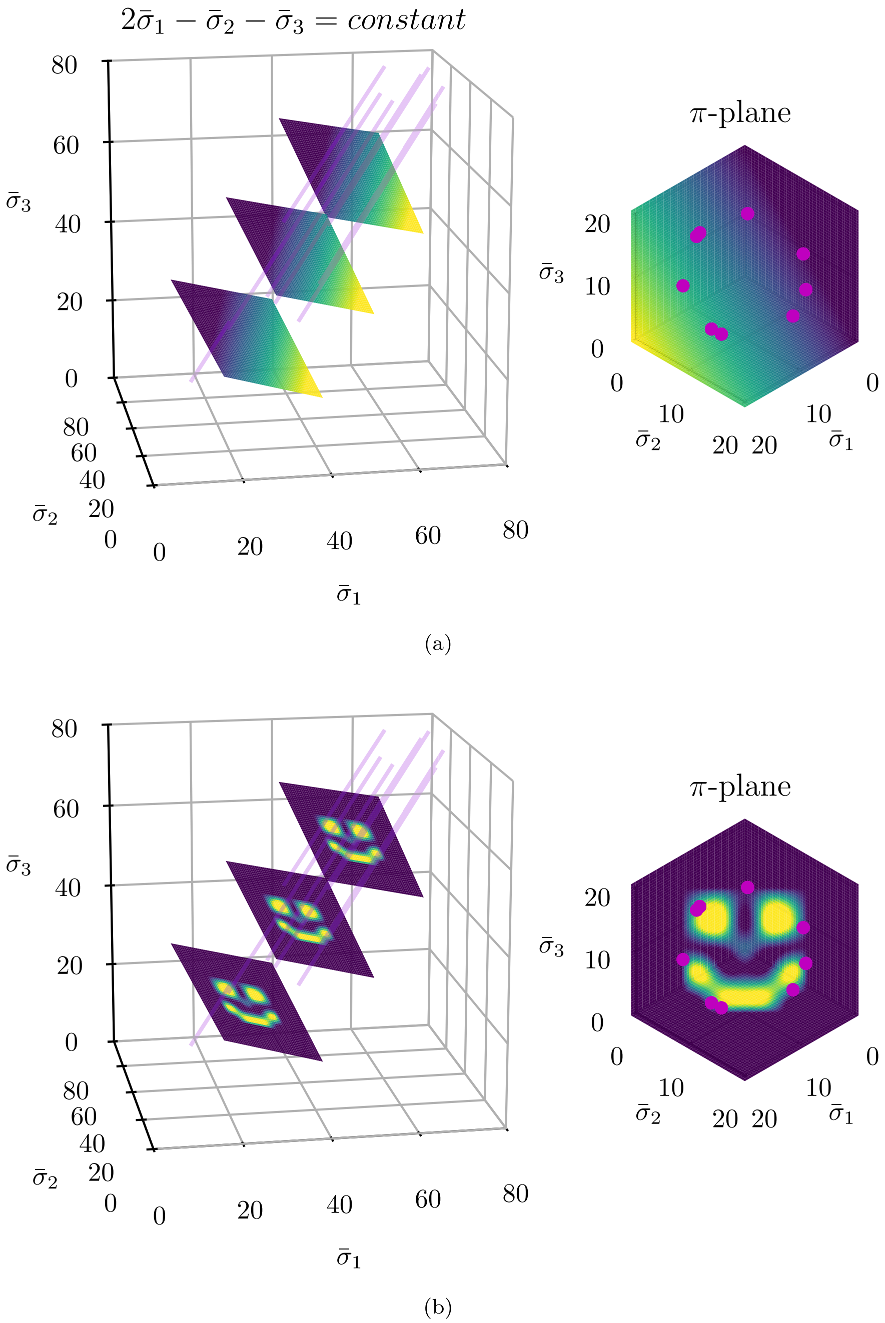


Fig. 4. Example of two trivial solutions of the SR problem defined by single stage loading of a von Mises MPE. The solutions are illustrated by 3 cross-sections which are perpendicular to the hydrostatic axis (also perpendicular to all load paths). Load paths are illustrated as purple lines (same data as Fig. 3). The trivial solutions are shown for (a) a simple, linear solution and (b) a happy face to show the arbitrariness of the potential parallel to the π-plane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

within 5% of the fitness value of the target (true) equation. In the more general circumstance where the target equation is not known *a priori*, convergence would instead need to be defined in a relative manner. In such a case, the use of a train/test split of the data could be used to quantify the performance of the resulting equations.

Average computation times for the 20 runs are reported, noting that the SR was run on dual socket 20 core 2.40 GHz Intel Gold 6148 Skylake (2×27.5 MB cache) processors. Because each core was associated with one genetic programming island, the total population size could be changed by using more or less cores.

4.1. Test case 1

Test case 1 illustrates the use of SR to derive an elastic-perfectly plastic von Mises yield surface from MPE response data. Forty cores (islands) were used for test case 1; fewer cores could have been used for this simple example but 40 represents the full utilization of one processor.

Fig. 7 shows the results for the 20 SR runs in test case 1. The fitness of the best equation of each of the 20 runs is illustrated as a function of GP generations, in Fig. 7(a). The extents of the boxplots

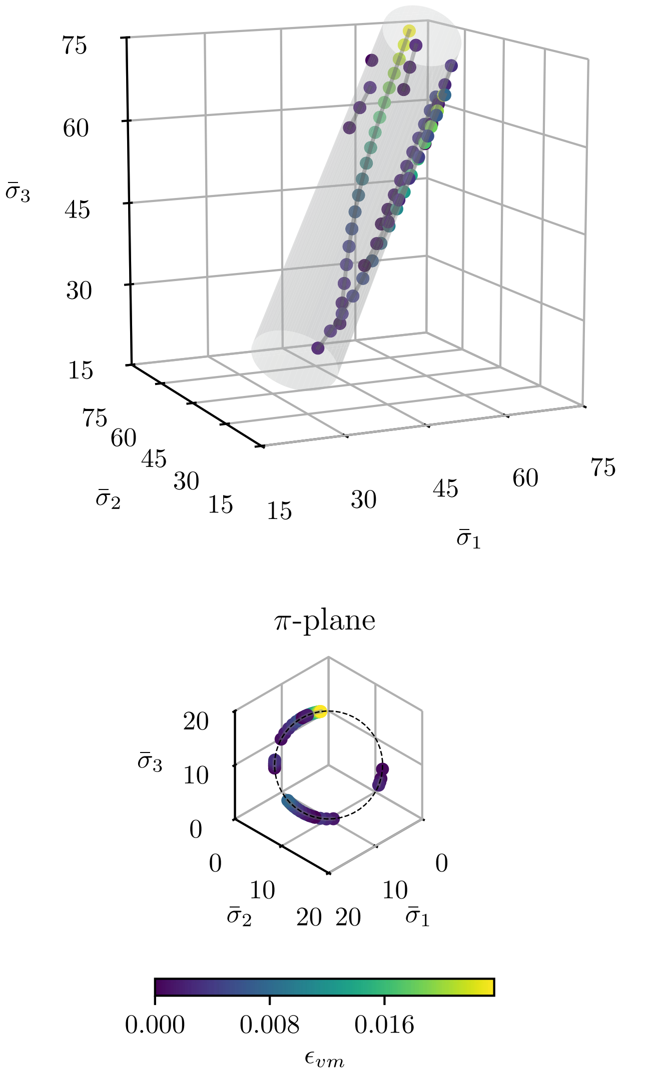


Fig. 5. The cylinder described by von Mises yield function without hardening and an exemplary subset of the dataset used in test case 1 (X_1).

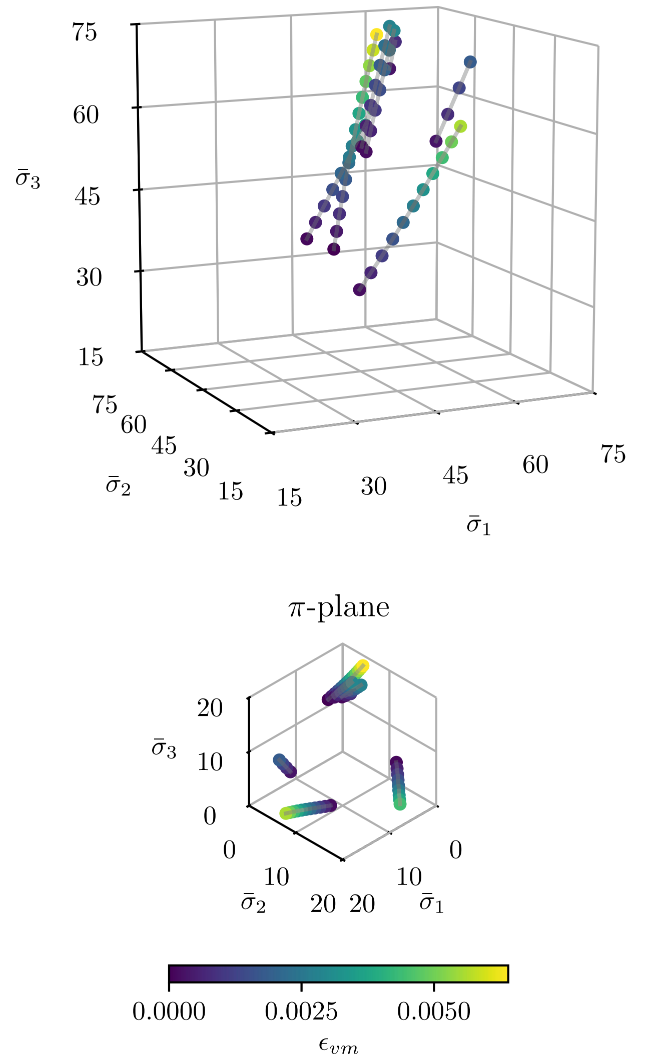


Fig. 6. An exemplary subset of the dataset used in test case 2 (X_2): von Mises yielding with linear hardening.

represent the innerquartile range with the purple line illustrating the mean. Outliers are shown as circles. The convergence criteria is illustrated as the dotted line to indicate the point at which a run is considered successful. Fig. 7(b) shows the percentage of the runs that have achieved convergence. Three observations should be made from the results. First, all 20 of the SR runs were successful in finding the exact functional form of the target equation. Second, the SR was able to effectively exclude the extraneous data ($\bar{\epsilon}_{vm}$) from all of the outputs. And third, all runs converged rapidly, the slowest taking only 850 generations. The average age to convergence was 417 generations and the average time to convergence was 31 s.

The output equation for one of the 20 runs is shown below. This particular equation was found after 400 generations.

$$\begin{aligned}&-43.22-((\bar{\sigma}_2-\bar{\sigma}_1)(\bar{\sigma}_3-\bar{\sigma}_1)+(\bar{\sigma}_3-\bar{\sigma}_2)(\bar{\sigma}_3-\bar{\sigma}_2))\\&=constant\end{aligned} \tag{24}$$

Rearanging the terms gives:

$$(\bar{\sigma}_2-\bar{\sigma}_1)(\bar{\sigma}_3-\bar{\sigma}_1)+(\bar{\sigma}_3-\bar{\sigma}_2)^2=constant. \tag{25}$$

With some algebra, this is seen to equate to Eq. (16).

The complexity of an equation is here defined as the number of nodes that describe the equation in acyclic graph form. For example, the complexity of the equation in Fig. 1 is 7. It is interesting to note that the result of the SR run shown in Eq. (24) has the same complexity as the target equation, despite the fact that it has an extraneous constant. After removing the constant, Eq. (24) has a complexity of 9 compared to the target equation's 11. For this run SR gives a succinct result, but such is not always the case. The equation below is the output from another one of the 20 SR runs; this equation was found in 100 generations.

$$\begin{aligned}&(\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1))(\bar{\sigma}_2-\bar{\sigma}_1\\&-(\bar{\sigma}_3-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1))-(\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1))\\&-(\bar{\sigma}_2-3.19-(\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1)+(\bar{\sigma}_2-\bar{\sigma}_1\\&-(\bar{\sigma}_3-\bar{\sigma}_1))(\bar{\sigma}_2-\bar{\sigma}_1)-(\bar{\sigma}_3-\bar{\sigma}_3)-(\bar{\sigma}_2-\bar{\sigma}_1\\&-(\bar{\sigma}_3-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1))\\&-(\bar{\sigma}_2-\bar{\sigma}_1)-((\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1))(\bar{\sigma}_2-\bar{\sigma}_1))\\&-((\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1))(\bar{\sigma}_2-\bar{\sigma}_1\\&-(\bar{\sigma}_3-\bar{\sigma}_1)-(\bar{\sigma}_2-\bar{\sigma}_1))))-(\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1)\\&-\bar{\sigma}_2)-(\bar{\sigma}_2-\bar{\sigma}_1)))+\bar{\sigma}_2-\bar{\sigma}_1-(\bar{\sigma}_3-\bar{\sigma}_1)\\&-(\bar{\sigma}_2-\bar{\sigma}_1)=constant\end{aligned} \tag{26}$$

With a complexity of 28, the above equation is a good example of what is known as bloat. In SR, bloat occurs when equations start to grow excessively large, often to a point where they become

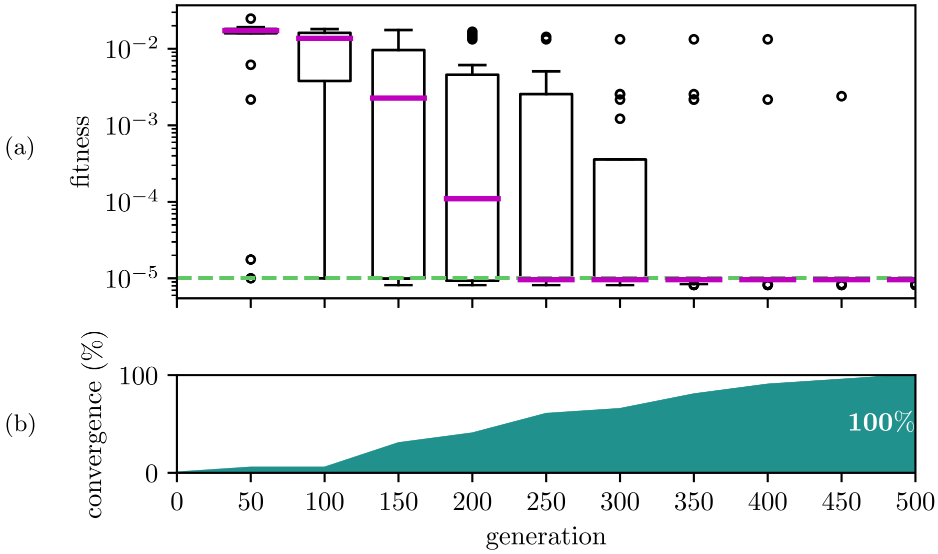


Fig. 7. The results of test case 1. (a) a box plot of the fitness of the most fit individuals from each of the 20 SR runs as a function of the number of GP generations. The median is illustrated by the purple line, and outliers are illustrated by circles. The convergence criteria is shown as the green dashed line. (b) the percentage of converged SR runs as a function of the number of generations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

unmanageable. In the run resulting in the above equation, the correct solution was found within 100 generations, in which case the bloat control measures (e.g., changing fitness predictors) may not have had time to become effective. Nevertheless, the above equation can be analytically simplified to:

$$3.19 + 2(\bar{\sigma}_1 - \bar{\sigma}_3)^2 + 2(\bar{\sigma}_2 - \bar{\sigma}_1)(\bar{\sigma}_2 - \bar{\sigma}_3) = constant, \quad (27)$$

which can be reorganized to look exactly as the target equation. The average complexity of the output equations for the 20 SR runs of test case 1 was 17, with 2 of the 20 runs returning the most succinct form of the target equation (complexity of 9).

4.2. Test case 2

Test case 2 describes a significantly more challenging problem for SR due to the presence of numerical constants. Because of the increased difficulty, 80 cores (islands) were used. The increased population size was found to be advantageous and lead to increased convergence rates; more details on the effect of population size to follow in Section 5.1.

Again, 20 SR runs are performed for this test case. The results for the 20 runs are illustrated in Fig. 8. In test case 2, an upper limit of 175,000 generations was set at which point any runs which did not converge were named unsuccessful. Only one of the 20 runs was unsuccessful. The vast majority of runs were successful and converged before 100,000 generations; a few outliers saw later convergence. The average age of convergence for the successful runs was 71,300 generations, and the average time to convergence for these runs was 2.36 h. However, average run times and ages for the successful runs may be misleading. A more insightful measure of performance includes the time and effort spent exploring the unsuccessful run. For this purpose, a measure of success performance (SP) age and SP time are adopted in likeness of the work of Kommenda et al. [27].

$$\text{SP age} = \frac{\text{total age of all runs}}{\text{number of successful runs}} \quad (28)$$

$$\text{SP time} = \frac{\text{total time spent on all runs}}{\text{number of successful runs}} \quad (29)$$

Generally, SP measures the average effort spent to achieve a successful run. The SP age for test case 2 was 80,500 generations and the SP time was 2.58 h.

The average complexity of the successful equations in test case 2 was 35. An example of a resulting equation with 35 complexity is shown below.

$$\begin{aligned}
&\epsilon_{vm} - (\bar{\sigma}_2 - \bar{\sigma}_3 - (\epsilon_{vm} + \bar{\sigma}_2 - \bar{\sigma}_1)) \\
&-((15.2)((-0.142)(-0.142)) + \epsilon_{vm} + (\epsilon_{vm} + \bar{\sigma}_2 \\
&-\bar{\sigma}_3 - (\epsilon_{vm} + \bar{\sigma}_2 - \bar{\sigma}_1))(\bar{\sigma}_2 - \bar{\sigma}_1) + \epsilon_{vm} - (-33.9) \\
&+((\epsilon_{vm} + (-0.142)(-0.142))(\epsilon_{vm} + \epsilon_{vm}))(490100) \qquad (30) \\
&+\epsilon_{vm} + \epsilon_{vm} - (\bar{\sigma}_2 - \bar{\sigma}_3 - (\epsilon_{vm} + \bar{\sigma}_2 - \bar{\sigma}_1) + \epsilon_{vm} \\
&+(-0.142)(-0.142) + (\bar{\sigma}_2 - \bar{\sigma}_3)(\bar{\sigma}_2 - \bar{\sigma}_3) - \epsilon_{vm})) \\
&= constant
\end{aligned}$$

While the above equation is complex, it can be simplified to:

$$\begin{aligned}
&\tfrac{1}{2}[(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_3 - \bar{\sigma}_1)^2 \\
&-c_1\epsilon_{vm} - c_2\epsilon_{vm}^2] = constant,
\end{aligned} \quad (31)$$

with $c_1 = 39596$ and $c_2 = 1960350$. Not only is this an equivalent form of the target equation, but after reorganization the constants c_1 and c_2 are both accurate to $\sim$ 0.0001% relative error.

The most succinct equation (i.e., least complex solution) found by SR in test case 2 was:

$$\begin{aligned}
&2.04 \times 10^{-06}\Big[((\bar{\sigma}_2 - \bar{\sigma}_3) + (\bar{\sigma}_1 - \bar{\sigma}_2))(\bar{\sigma}_2 - \bar{\sigma}_3) \\
&+(\bar{\sigma}_1 - \bar{\sigma}_2)^2\Big] - \epsilon_{vm}(\epsilon_{vm} + 0.0404 + \epsilon_{vm}) = constant,
\end{aligned} \quad (32)$$

which has a complexity of 18. Though this may appear complex, the above equation is actually less complex than the target equation as written in Eq. (20), which has a complexity of 19. It is also nearly the most succinct representation of the target equation (complexity 17).

The high success rate in test case 2 shows that SR was able to reliably find the target plastic yield equation despite the inclusion of numerical constants.

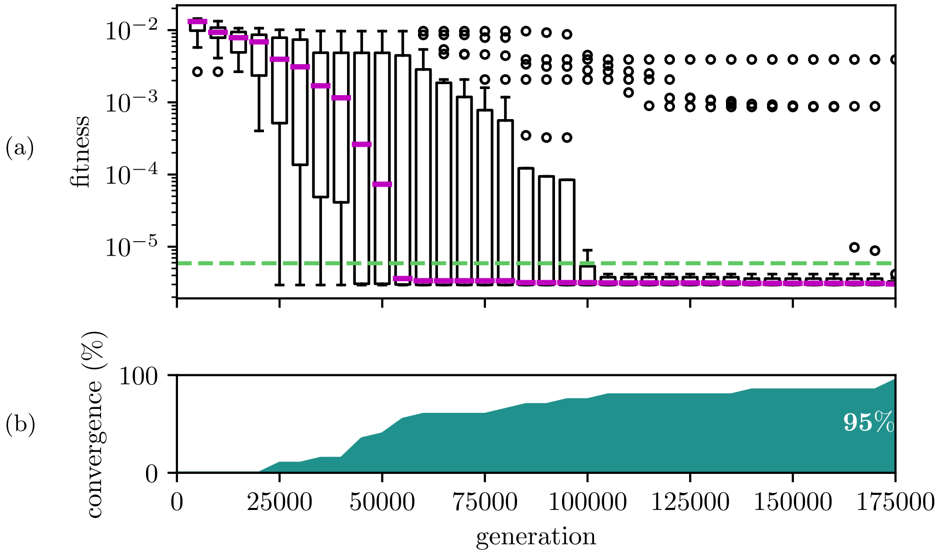


Fig. 8. The results of test case 2. (a) a box plot of the fitness of the most fit individuals in each of the 20 SR runs as a function of the number of generations. The median is illustrated by the purple line, and outliers are illustrated by circles. The convergence criteria is shown as the green dashed line. (b) the percentage of converged SR runs as a function of the number of generations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.3. Test case 3

Test case 3 centered on the search for an internal variable evolution equation in a von Mises plasticity model with linear hardening. It provides the complement to test case 2; their combination produces a complete plasticity model. Forty cores (islands) were used for test case 3. As in test case 1, fewer cores could have been used, but 40 fully utilized one processor.

Twenty SR runs were executed for test case 3 using the dataset X_3 as described in Eq. (21). All of the runs finished within a few seconds, and had some form of the following equation as the result.

$$\Delta\bar{\epsilon}_{p1} + \Delta\bar{\epsilon}_{p2} + \Delta\bar{\epsilon}_{p3} = constant = 0 \tag{33}$$

The data in dataset X_3 can be substituted into the equation to give the value of 0 for the constant. Although the above equation is not the target equation it does represent an accurate physical description of the dataset. In this case, SR was able to identify that the plastic deformation is deviatoric, without explicitly targeting that equation.

To proceed in finding the variable evolution equation the dataset for test case 3 was modified by removing $\Delta\bar{\epsilon}_{p3}$.

$$X_{3'} = \left\{ \Delta\bar{\epsilon}_{p1}^{(ij')}, \Delta\bar{\epsilon}_{p2}^{(ij')}, \Delta\bar{\epsilon}_{vm}^{(ij')} \right\} \tag{34}$$

The removal of $\Delta\bar{\epsilon}_{p3}$ ensures that the deviatoric plastic strain equation (Eq. (33)) cannot be found again by SR. Similarly, removing $\Delta\bar{\epsilon}_{p3}$ from the target equation by substituting the value of $\Delta\bar{\epsilon}_{p3} = -\Delta\bar{\epsilon}_{p1} - \Delta\bar{\epsilon}_{p2}$ from Eq. (33) into Eq. (23) gives the new target equation.

$$\Delta\bar{\epsilon}_{p1}^2 + \Delta\bar{\epsilon}_{p1}\Delta\bar{\epsilon}_{p2} + \Delta\bar{\epsilon}_{p2}^2 - \frac{3}{4}\Delta\bar{\epsilon}_{vm}^2 = constant \tag{35}$$

The result of 20 SR runs using the modified dataset $X_{3'}$ is shown in Fig. 9. All 20 runs successfully converged within 3000 generations with an average convergence age of 1615 generations and an average convergence time of 127 s. The average complexity of the resulting equations was 29 with the most succinct result having a complexity of 17 (the simplest form of the target equation has a complexity of 12).

The higher value of the convergence criteria is of particular interest in test case 3. The convergence value here is more than an order of magnitude larger than the previous two test cases. The increased value is due to the inherent noise in the input dataset (noise stemming from inaccuracies in derivatives). With the presence of noise, some amount of misfitting did occur. For example, the following equation is one result with average complexity (analytically simplified from its original form):

$$\begin{aligned}
&\Delta\bar{\epsilon}_{vm} - \Delta\bar{\epsilon}_{vm}(1.071 + 2\Delta\bar{\epsilon}_{vm})\\
&+(2.36 * 10^{-}6[-3.41 * 10^6 - 2\Delta\bar{\epsilon}_{p1} - 3\Delta\bar{\epsilon}_{p2}]\\
&*(-\Delta\bar{\epsilon}_{p1}\Delta\bar{\epsilon}_{p2} + (\Delta\bar{\epsilon}_{p1} + \Delta\bar{\epsilon}_{p2} - \Delta\bar{\epsilon}_{vm})\\
&*(\Delta\bar{\epsilon}_{p1} + \Delta\bar{\epsilon}_{p2} + \Delta\bar{\epsilon}_{vm}))) = constant.
\end{aligned} \tag{36}$$

The term in brackets on the second line of the equation can be approximated $[-3.41 * 10^6 - 2\Delta\bar{\epsilon}_{p1} - 3\Delta\bar{\epsilon}_{p2}] \approx -3.41 * 10^6$. Similarly, the first line of the equation can be approximated $\Delta\bar{\epsilon}_{vm} - \Delta\bar{\epsilon}_{vm}(1.071 + 2\Delta\bar{\epsilon}_{vm}) \approx -2\Delta\bar{\epsilon}_{vm}^2$. With these two easily identifiable approximations the above equation simplifies to:

$$\begin{aligned}
&-8.046(\Delta\bar{\epsilon}_{p1}^2 + \Delta\bar{\epsilon}_{p1}\Delta\bar{\epsilon}_{p2} + \Delta\bar{\epsilon}_{p2}^2 - 0.7514\Delta\bar{\epsilon}_{vm}^2)\\
&= constant.
\end{aligned} \tag{37}$$

This is equivalent to the target equation with a relative error on the $\Delta\bar{\epsilon}_{vm}^2$ constant of 0.19%. Because the unsimplified version of this equation had a better fitness than the target equation, the added complexity is attributed to overfitting of the dataset. A post-processing script was developed to numerically simplify the result equations. The script randomly prunes branches of the acyclic graph and keeps the modification if the fitness function does not change significantly. Running the script on the outputs of the 20 runs ensured that the correct functional form of the target equation could be recovered. It should be noted that other overfitting mitigation techniques, e.g., cross-validation or fitness-complexity Pareto analysis, could be used to achieve similar results.

5. Robustness and efficiency of test cases

Application of the proposed approach for plasticity model development on higher fidelity material RVEs and more complex yield functions will involve increased computational costs. For this reason, it is important to investigate how the robustness and efficiency of the SR-based approach are affected by computational

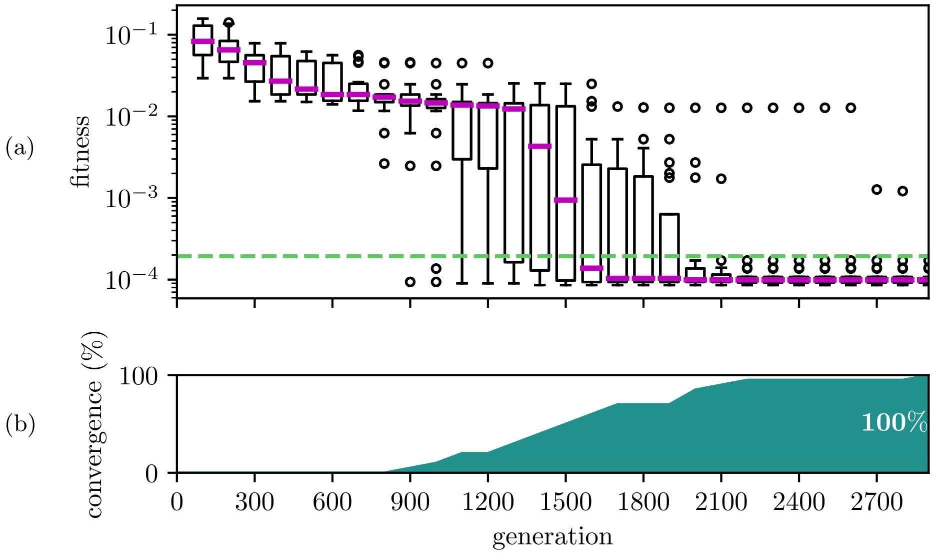


Fig. 9. The results of test case 3. (a) a box plot of the fitness of the most fit individuals in each of the 20 SR runs as a function of the number of generations. The median is illustrated by the purple line, and outliers are illustrated by circles. The convergence criteria is shown as the green dashed line. (b) the percentage of converged SR runs as a function of the number of generations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

resources and expected model complexity. The following sections include studies of such relationships.

5.1. Effect of GP population size

Population size has a significant effect on GP [50,51]. Using test case 2 as an example problem, the effect of population size was investigated. The population size was varied in a number of SR runs by increasing the number of GP islands, maintaining one processing core per island and 128 solution equations per island. Twenty SR runs (or as many as could be run in a 72 h time limit) were executed for each investigated number of cores, ranging from 40 to 160 in increments of 10.

The effect of population size on success rate is shown in Fig. 10. Consistent with examples in the work of Samples et al. [51], higher success rates are seen with increasing population size. The higher success rates are due to both the increased diversity and computational effort that comes with a larger population. A drastic increase in success rate with population size is seen up until about 80 islands, after which the increase is more gradual. This suggests that there may be some minimum population size at which convergence becomes more consistent. Both SP age and SP time decrease with increasing population sizes, both decreasing at about the same rate: see Fig. 11(a) and (b). SP time is a measure of wall clock time, but when considering computational efficiency total CPU-time is a better metric for the computational expense. Fig. 11(c) illustrates that the total CPU-time is more noisy. A rough trend may be gleaned from Fig. 11(c), i.e., that there is an apparent minimum around 80 islands with a gradual increase with larger populations thereafter. This trend is linked to the success rate. With fewer than 80 islands, success rates rise rapidly with increasing number of islands, causing a drop in SP CPU-time. With more than 80 islands, success rates are more consistent. The cause of the gradual increase in SP CPU-time after 80 islands is less apparent. The trend of SP island-generations with respect to population size (Fig. 11(d)) shows that the increasing SP CPU-time after 80 islands is accompanied by increasing SP island-generations. This may indicate that the increased time for gene mixing to occur (caused by having more islands) becomes more important with larger population sizes.

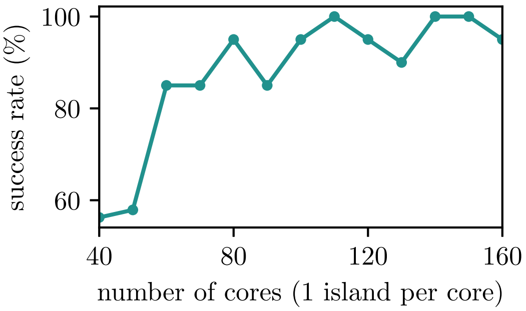


Fig. 10. The effect of population size on the success rate of test case 2.

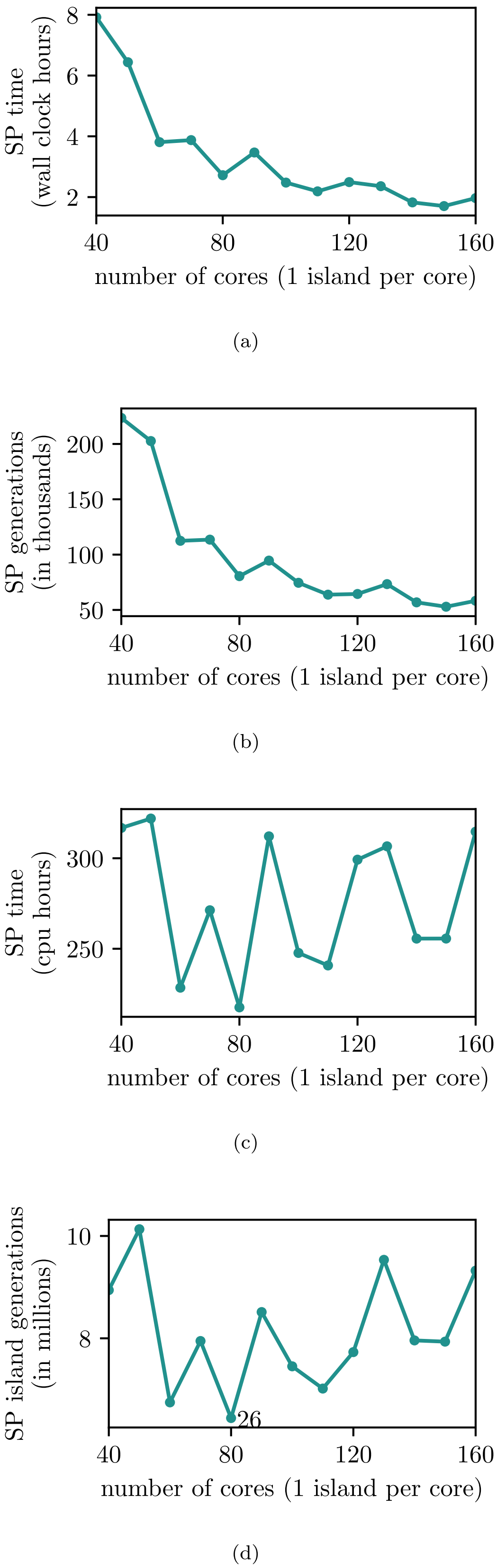
5.2. Effect of dataset size

Unlike the simple test cases, real material systems often have complicated microstructures. Simulation of complicated microstructural RVEs requires significant computational power. For this reason, it is important to consider the amount of computational time required to generate the necessary datasets. The computational cost associated with generating the datasets is proportional to the product $(n_L)(n_s)$, assuming that each RVE load step has approximately the same cost. Thus, the ideal choice for n_L and n_s are the minimum values for which an accurate model can be found using SR.

A study was performed to illuminate the effect of input dataset size on the efficiency of SR identification of plastic yield functions. Test case 1 was repeated numerous times, with all combinations of

$$\begin{aligned} n_s \in \; & \{16, 18, 20, 22, 25, 28, 33, \\ & 40, 50, 66, 100, 200, 400\} \\ n_L \in \; & \{40, 45, 50, \ldots, 100\}. \end{aligned} \tag{38}$$

Each combination of n_L and n_s was used in 20 SR runs using 40 islands/cores. The average convergence time (equivalent to SP time here because all simulations were successful) is shown as a function of n_s and n_L in Fig. 12. In Fig. 12(a), a bowl shaped minimum is seen around the region of $n_s = 28, 33$, making these the ideal choices for n_s. The region to the left of the minimum corresponds to data-



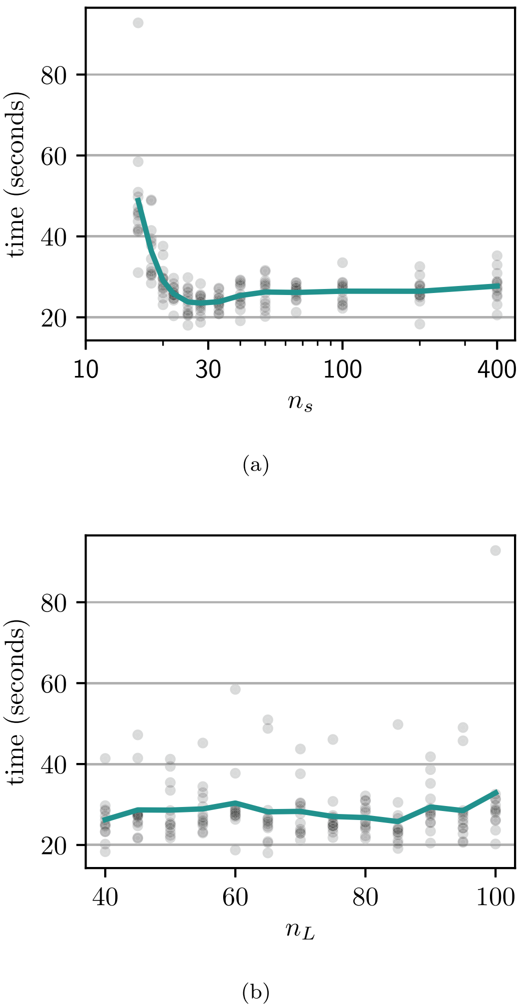
(a)

(b)

(c)

(d)

Fig. 11. The effect of population size on the efficiency of SR in test case 2.



(a)

(b)

Fig. 12. The effect of dataset size on convergence time of SR in test case 1. Each circle represents one combination of n_L and n_s from the sets defined in Eq. (38). The lines represent average values.

starved SR, where more data gives both more accurate derivative calculations and more information about the yield surface, and thus, leads to more efficient convergence. The region to the right of the minimum corresponds to a data-glut where the computational burden of additional data outweighs its utility. The value of the ideal n_s is difficult to know *a priori* because, presumably, it is dependent upon the complexity of the underlying yield surface. However, because test case 1 represents one of the simplest yield surfaces, a value of $n_s = 28$ represents a good lower limit for more complex material systems. In the cases of difficult FE simulations requiring numerous time steps, downsampling the time steps is a potentially avenue for improving the speed of SR.

Convergence time is relatively invariant with respect to the number of loading cases n_L (Fig. 12(b)). An investigation was performed to find the limit of the minimum n_L for which convergence time remained invariant, utilizing $n_s = 28$. It was found that using less than 15 loading cases resulted in increased convergence times. Using the same justification as above, $n_L = 15$ represents a good lower limit on the number of loading cases that should be simulated for more complex material systems.

5.3. Effect of plasticity model complexity

Generally, increasing complexity of the targeted models causes an increase in the computational cost required by SR to discover those models [18,19]. Increased complexity can come in many forms. The three forms covered by the above test cases are (1)

Table 2
Complexity measures of the test cases.

Test case	1	2	3
Comp. effort (SP-CPU hours)	0.35	208	1.41
Complexity of form	11	17	12
Variable dimension	3	4	3
Constant dimension	0	2	1

the complexity of the target functional form, (2) the variable dimension of the target, and (3) the number of required numerical constants. The computational effort required for each of the test cases is shown in Table 2 along with these three complexity measures.

A significant increase in computational effort (about 600x) was required in test case 2 compared to test case 1. Insight into how that 600x cost increase is distributed into contributions from the above three components of complexity can be found from Fig. 8–4 in reference [19] and Fig. 9 in reference [18]. Roughly a 10x increase is attributed to increased size of the functional form, and about a 1.2x increase is attributed to increased variable dimension. The remaining 50x increase is attributed to the addition of two numerical constants. While the above breakdown may be used for quick assessment, the true dependence of computational cost on model complexity is itself complex and nonlinear.

A significant decrease in computational effort was required in test case 3 with respect to test case 2. This is due in part to the looser tolerance for convergence; however, by inspection of Fig. 8 we can limit this effect to at most a factor of 2.

When considering scaling to more complex materials systems, the increase in computational cost for SR could be concerning; however, the proportion of time spent on SR compared to data generation with complex RVEs is small. Additionally, the building of a SR model could be thought of in an iterative fashion that incrementally introduces complexity. In this case, introduction of knowledge from prior SR runs could be incorporated in successive training, e.g., in the manner of reference [52]. With all of this in mind, the scaling of computational cost associated with SR of more complex yield surfaces is not currently seen as a prohibitive limitation of the method.

6. Application to porous plasticity

The above approach is applied to the development of a constitutive model for porous plasticity in this section. This application represents a realistic scenario characterized by (1) the generation of response data from an inhomogeneous RVE, (2) a much more complex system, and (3) a yield function without an expected true result. The results of the current method are compared to two well-known porous plasticity models, the Gurson model [53] and the Gurson-Tvergaard-Needleman (GTN) model [54]. These two models can be described by the yield function:

$$\Phi = \left(\frac{\sigma_{vm}}{\sigma_y}\right)^2 + 2q_1 v \cosh\left(\frac{3}{2}q_2\frac{\sigma_h}{\sigma_y}\right) - \left(1 + (q_1 v)^2\right), \tag{39}$$

where v is void volume fraction, σ_{vm} is the von Mises stress, σ_h is the hydrostatic stress, q_i are parameters, and σ_y is the yield stress.. The case where $q_1 = q_2 = 1$ is the Gurson model. The case where $q_1 = 1.25$ and $q_2 = 1$ is referred to as the GTN model in this work, based on recommendations from that original work. Note that there are other differences between the GTN and Gurson models, such as GTN's use of an effective void volume fraction to simulate void interaction, but the current porous plasticity example will be restricted to a regime where the change in q_i is the primary difference between the models.

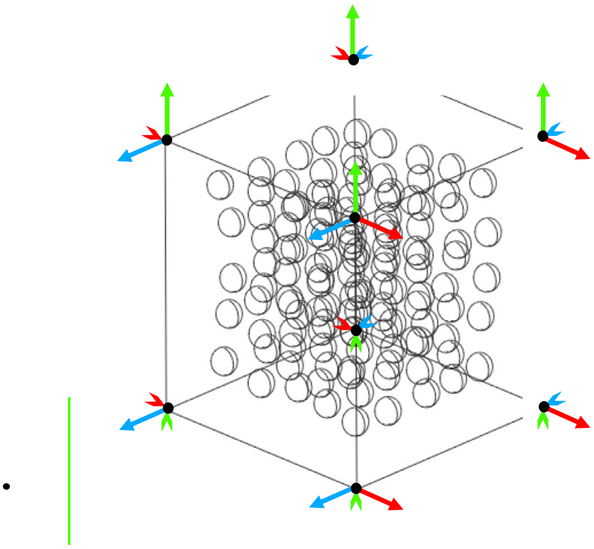


Fig. 13. RVE containing voids with arrows depicting the loading directions. Three faces have fixed boundary conditions while the other three faces have displacement boundary conditions applied perpendicular to the plane.

A finite element model of a unit cube with spherical voids was created to model an RVE of a porous material. The RVE includes 125 voids with the void volume fraction and void diameter chosen as 0.0632 and 0.0985, respectively. The voids are placed in the cube by initially organizing the voids on a grid and then randomly perturbing each void. Each void is locally perturbed in each direction by a random amount up to 20% of a void diameter. A mesh convergence study was performed leading to a choice of a mesh seed size of 0.1. The use of hybrid quadratic tetrahedral elements was chosen to combat issues with incompressibility and ill-conditioning. Additionally, non-linear geometry was used during the simulation to account for the effects of void growth. The material parameters in test case 1, an elastic-perfectly plastic material, are used to model the matrix of the RVE.

The RVE was loaded through displacement boundary conditions on the faces of the cube, as illustrated in Fig. 13. It was subjected to two-stage loading, as in Eq. (14), with the second loading stage as a permutation of the first stage. The second loading stage was applied once the volume-averaged equivalent plastic strain exceeded 10^{-4}. $n_L = 169$ different loading cases, defined by varying load ratios f_1,f_2, and f_3, were chosen using Latin hypercube sampling so that data would be approximately evenly distributed in stress space. A maximum applied displacement of 0.006 was chosen through preliminary simulations of uniaxial and hydrostatic loading cases with the intention of limiting interaction between voids. The interaction of voids was limited firstly for performance reasons, but also with the intention of remaining in a physical domain where both the Gurson and GTN model are relevant. An additional convergence study was conducted to determine $n_s = 145$ increments. Both n_L and n_s are well above the minimum values suggested in Section 5.2.

The volume-averaged response data from the plastic portion of the second loading stage is used as the training data for SR. Inspired by the Gurson and GTN models, the volume averaged principle stresses are transformed into a cylindrical coordinate system, i.e.

$$\{\bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3\} \rightarrow \{\bar{\sigma}_h, \bar{\sigma}_{vm}, \bar{L}\}. \tag{40}$$

The transformation is defined by:

$$\begin{aligned} \bar{\sigma}_h &= \tfrac{\bar{I}_1}{3} \\ \bar{\sigma}_{vm} &= \sqrt{3\bar{J}_2} \\ \bar{L} &= \tfrac{3\sqrt{3}}{2}\tfrac{\bar{J}_3}{\bar{J}_2^{\frac{3}{2}}}, \end{aligned} \tag{41}$$

where

$$\begin{aligned} \bar{I}_1 &= \bar{\sigma}_1 + \bar{\sigma}_2 + \bar{\sigma}_3 \\ \bar{J}_2 &= \tfrac{1}{6}\left[(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_3 - \bar{\sigma}_1)^2\right] \\ \bar{J}_3 &= (\bar{\sigma}_1 - \bar{\sigma}_h)(\bar{\sigma}_2 - \bar{\sigma}_h)(\bar{\sigma}_3 - \bar{\sigma}_h). \end{aligned} \tag{42}$$

Here, $\bar{L}$ is the volume-averaged Lode parameter, which has been shown to be an effective parameter for quantifying porous plastic behavior [35,36,55].

All GP runs in this section were performed using the same set of hyperparameters as the test cases in the previous sections (see Table 1), and were repeated 5 times. Based on the results of Section 5.1, 160 islands (cores) were used to give an increased chance of success. The runs were limited to 500,000 generations or 72 h wall-clock time, whichever occurred first.

The identification of the plastic yield function was pursued using a training dataset that consisted of the volume-averaged stresses, the volume averaged equivalent plastic strain, and the void volume fraction.

$$X_4 = \left\{\bar{\sigma}_h^{(ij')}, \bar{\sigma}_{vm}^{(ij')}, \bar{L}^{(ij')}, \bar{\epsilon}_{vm}^{(ij')}, \bar{v}^{(ij')}\right\} \tag{43}$$

This data was limited to the portion of the second loading stage where plasticity was actively occurring, indicated above by the superscripts ij'. GP was performed for the above dataset, and early in the evolution process a relationship was identified:

$$\bar{v} - \bar{\epsilon}_{vm}\left(c_1\bar{\sigma}_{vm} + c_2\bar{\sigma}_{vm}^3 + c_3\bar{\sigma}_{vm}\bar{\epsilon}_{vm} + c_4\right) = constant, \tag{44}$$

with $c_1 = 0.299, c_2 = -0.811, c_3 = -13.5$, and $c_4 = 0.622$. A similar relationship was found in all 5 of the GP runs. Notably, this is not a yield function but rather a relationship between a subset of the parameters; this is a direct analogue to the identification of deviatoric plastic strain in test case 3. The identification of hidden relationships, such as this one, using SR could be useful in feature engineering or reduced order modeling. Here, the above relationship can be reorganized and viewed as a predictive relationship for void volume fraction:

$$\bar{v} = c_0 + \bar{\epsilon}_{vm}\left(c_1\bar{\sigma}_{vm} + c_2\bar{\sigma}_{vm}^3 + c_3\bar{\sigma}_{vm}\bar{\epsilon}_{vm} + c_4\right), \tag{45}$$

where the value $c_0 = 0.0628$ was chosen as the mean of Eq. (44) evaluated over the entire dataset. By setting $\bar{\epsilon}_{vm} = 0$ it can be seen that the initial void volume fraction is estimated by c_0, which is within 0.6% of the initial void volume fraction of the RVE. As illustrated in Fig. 14 the predictive accuracy of the equation is high; $r^2 = 0.995$ for the entire dataset. In the hydrostatic limit of the equation (i.e., $\bar{\sigma}_{vm} = 0$), void volume fraction is a linear function of equivalent plastic strain. Within the range of the variables in the dataset, it is also observed that void volume fraction growth is slowed by increasing $\bar{\sigma}_{vm}$. This is a property that matches well with the general trends seen in the literature, and corresponds to the intuition that less triaxially dominant loadings promote more void deformation/shearing rather than void growth.

As in test case 3, the identification of a yield function can proceed with the removal of one of the parameters from the dataset. For similarity with the Gurson/GTN yield function, which contain void volume fraction and von Mises stress, equivalent plastic strain was removed from the dataset.

$$X_{4'} = \left\{\bar{\sigma}_h^{(ij')}, \bar{\sigma}_{vm}^{(ij')}, \bar{L}^{(ij')}, \bar{v}^{(ij')}\right\} \tag{46}$$

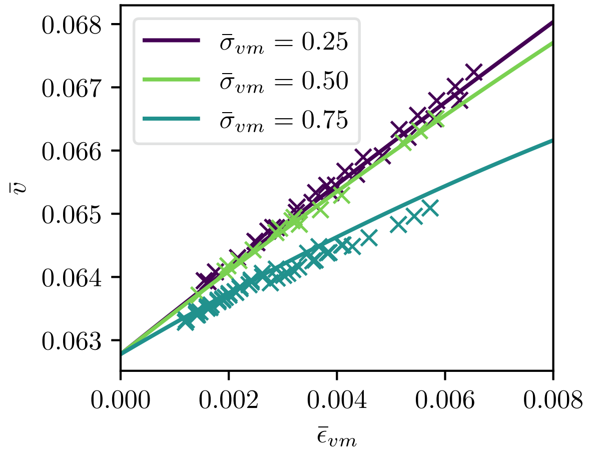


Fig. 14. Relationship of void volume fraction to equivalent plastic strain and von Mises stress. Eq. (45) is illustrated as solid lines. Data in X_4 that lie within a threshold of 0.01 of the $\bar{\sigma}_{vm}$ values are illustrated as x-markers.

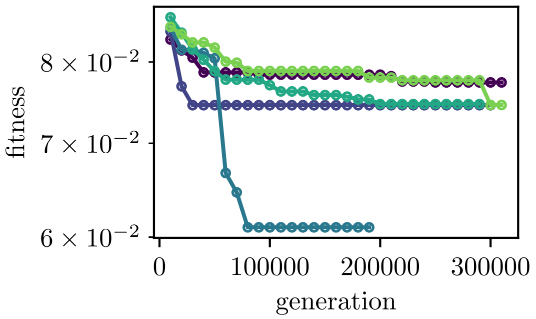


Fig. 15. The results of the porous plastic yield function SR. The fitness is shown for the most fit individuals in each of the 5 SR runs.

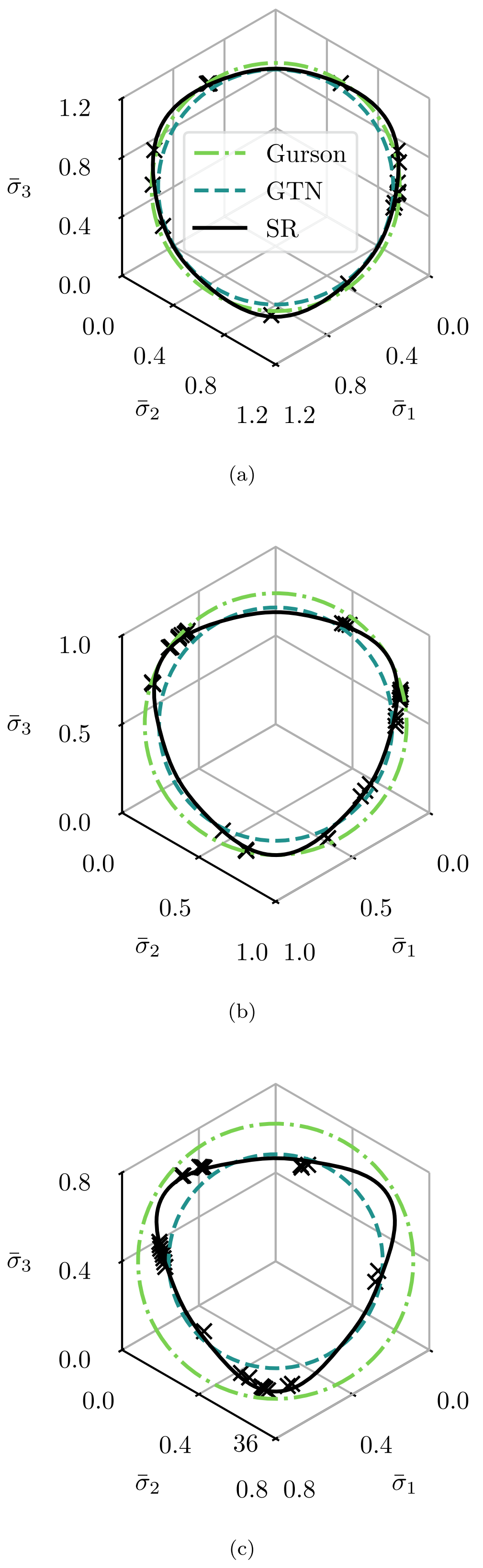
The results of using $X_{4'}$ in five SR runs are illustrated in Fig. 15. One of the five runs was able to produce a result with lower fitness than the others; this could be an indication that there was some limiting factor in SR for the evolution in this case. For instance, this could be a limitation on the number of generations, as a similar result would have been seen if test case 2 were limited to 30,000 generations. Another possibility, is that premature convergence is occurring in GP. The standard approach to mitigating premature convergence in GP is through the use of multiple restarts of the same analysis. Viewing the five SR runs in this manner, the remainder of this subsection focuses on the best equation encountered across all SR runs

The yield equation with the lowest fitness is:

$$\begin{aligned} &\left[\bar{\sigma}_{vm}(4\bar{\sigma}_h^2 + (2\bar{v} - \bar{L} + c_1)\bar{\sigma}_h + c_2) - \bar{\sigma}_h(c_3\bar{v}^2 + c_4)\right] \\ &\cdot(c_5\bar{v}^2 + 2\bar{v} + c_6) \\ &+c_7\bar{v}^2 - \bar{\sigma}_{vm}\bar{L}^2(c_8\bar{v}^2 - c_9) + c_{10} = constant, \end{aligned} \tag{47}$$

with c_1 through c_{10} equal to $7.47, -33.75, 843, 20.0, -5.24 \times 10^9, 2.04 \times 10^7, -2.23 \times 10^{11}, 2.62 \times 10^9, 1.02 \times 10^7$, and 8.83×10^3, respectively. Note that the equation has been simplified and reorganized from its GP representation. Setting the constant equal to its mean value over $X_{4'}, c_{11} = -8.67 \times 10^8$, the yield function is then defined:

$$\begin{aligned} \Phi_{SR} = \quad &\left[\bar{\sigma}_{vm}(4\bar{\sigma}_h^2 + (2\bar{v} - \bar{L} + c_1)\bar{\sigma}_h + c_2)\right. \\ &\left.-\bar{\sigma}_h(c_3\bar{v}^2 + c_4)\right](c_5\bar{v}^2 + 2\bar{v} + c_6) \\ &+c_7\bar{v}^2 - \bar{\sigma}_{vm}\bar{L}^2(c_8\bar{v}^2 - c_9) + c_{10} - c_{11} = 0. \end{aligned} \tag{48}$$



(a)

(b)

(c)

Fig. 16. Level sets of the porous plasticity yield equation produced by SR (Eq. 48) projected onto the π-plane. The levels are (a) $\bar{\sigma}_h = 1.0, \bar{v} = 0.0635$; (b) $\bar{\sigma}_h = 1.3, \bar{v} = 0.064$; and (c) $\bar{\sigma}_h = 1.5, \bar{v} = 0.0645$. The green and teal lines indicate the level sets of the Gurson and GTN equations, respectively. Data in $X_{4'}$ that lie within a threshold of 0.01 of the $\bar{\sigma}_{vm}$ values and 0.001 of the $\bar{v}$ values are illustrated as x-markers.

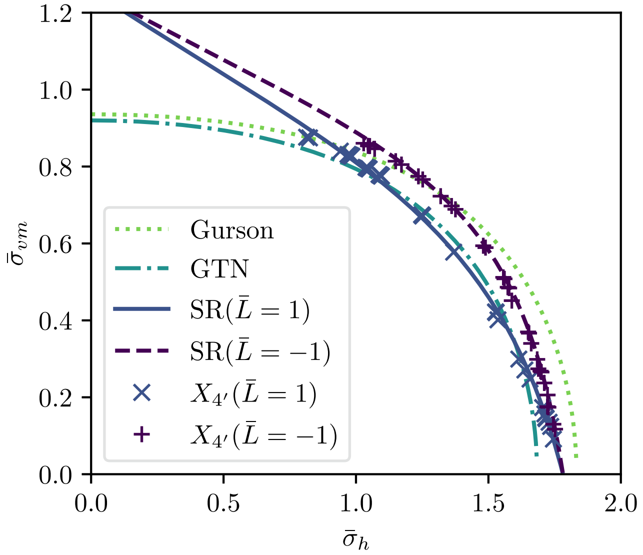


Fig. 17. Pressure dependence of the porous plasticity yield equation produced by SR (Eq. (48)). All yield functions are shown for $\bar{v} = 0.064$. Data in $X_{4'}$ that lie within a threshold of 0.005 of the $\bar{L}$ values and 0.001 of the $\bar{v}$ value are illustrated as x- and +-markers.

This yield function has both a higher complexity and a higher constant dimension than all the test cases, which is likely the cause of the greater computation time required to obtain the equation with SR. Note that the above yield function does not contain hyperbolic cosine terms that are present in the Gurson/GTN yield function. This is because the mathematical operators were limited (as in all test cases) to addition, subtraction, and mutliplication.

One of the advantages of SR is the interpretabilty of the produced models. In this case, it is possible to glean some important aspects of the model from the investigation of its form. Firstly, the limit of the yield function as $\bar{v} \rightarrow 0$ describes the response of a fully-dense material; the known true response of the fully-dense material is equivalent to that of its von Mises matrix. The SR yield function, however, does not model this asymptotic behavior since $\Phi_{SR}(\bar{v} = 0)$ has a dependence upon $\bar{\sigma}_h$ and $\bar{L}$:

$$\begin{aligned}\Phi_{SR}(\bar{v} = 0) = \;& c_6\left[\bar{\sigma}_{vm}(4\bar{\sigma}_h^2 + (-\bar{L} + c_1)\bar{\sigma}_h + c_2) - c_4\bar{\sigma}_h\right] \\ & + c_9\bar{\sigma}_{vm}\bar{L}^2 + c_{10} - c_{11} = 0.\end{aligned} \tag{49}$$

Secondly, the hydrostatic response of the yield function is described by

$$\begin{aligned}\Phi_{SR}(\bar{\sigma}_{vm} = 0) = \;& -\bar{\sigma}_h(c_3\bar{v}^2 + c_4)(c_5\bar{v}^2 + 2\bar{v} + c_6) \\ & + c_7\bar{v}^2 + c_{10} - c_{11} = 0\end{aligned} \tag{50}$$

and resembles the form of the GTN yield function in that it contains only terms for $\bar{\sigma}_h$ and $\bar{v}$. Notably, this function does not depend on $\bar{L}$ which is intuitive because as stress becomes purely hydrostatic, the difference between $\bar{L} = 1$ and $\bar{L} = -1$ approaches zero.

It is worth noting that the limiting behaviors associated with Eqs. (49) and (50) are both outside the range of the training data set $X_{4'}$. If these limiting behaviors are of high importance, then additional data could be included in the SR runs to increase accuracy in these regimes. For instance, the data contained in X_1 (after appropriately transforming and adding $\bar{v} = 0$) could be added to $X_{4'}$ so that the fully-dense material behavior would be represented. Care must be taken, though, to have a balanced dataset so that the fitting of one regime is not favored over the other.

The stress dependence of the SR yield function is illustrated as projections onto the π-plane in Fig. 17 for fixed values of $\bar{\sigma}_h$ and $\bar{v}$. The yield equation is compared to both the Gurson and GTN yield equations. It is seen that the SR yield function is very similar to the Gurson and GTN functions for lower values of $\bar{\sigma}_h$ in Fig. 16 (a). With increased values of $\bar{\sigma}_h$ and $\bar{v}$, the shape of the SR surface changes from circular(matching Gurson and GTN) to more triangular. Qualitatively, yield functions of this shape have been seen in the literature for porous materials [e.g., 36]. More importantly, the data in $X_{4'}$ (illustrated with x-markers in Fig. 17) clearly show that the response of the RVE is better captured with the SR yield equation than the circular Gurson and GTN equations.

The pressure dependence of the SR yield function is illustrated in Fig. 17 for a fixed value of $\bar{v} = 0.064$. In this case, it is seen that the shape of the Gurson and GTN models are reasonable approximations of the data associated with the RVE. The critical exception here is their lack of dependence upon $\bar{L}$, which the SR model more accurately captures. The convergence of the $\bar{L}$ response as $\bar{\sigma}_{vm} \rightarrow 0$ (discussed above), can also be seen in Fig. 17; it represents known physics of the system. Additionally, it is seen in Fig. 17 that the SR yield function performs nonphysically in the region $\bar{\sigma}_h \lessapprox 0.8$. This behavior could be remedied with additional data in that region.

7. Conclusions

A framework has been presented in this work for the use of symbolic regression (SR) in the development of interpretable, data-driven plasticity models. The framework is a type of automatic computational homogenization wherein load-response data from RVE simulations of a material system are used to develop a homogenized model with SR. The framework was successful in three test cases, delivering equations which are readily interpretable in an automatic fashion. The successful test cases illustrated the capacity of SR to be used in plasticity model development with increasing complexity and noisiness. Of particular note is the ability of SR to produce accurate models containing real-valued numerical constants. This previous challenge for SR, has been overcome here by a combination of local optimization and increased compute power.

The framework was also applied to the identification of a porous plastic yield function where it was shown to result in a model that accurately described the response of a representative volume element. The results of the porous plasticity application illustrated two important aspects of the framework. First, the produced model is most accurate in regions with data, and becomes less accurate with extrapolation. Secondly, the interpretable nature of the model allowed for the assessment of model behavior, such as asymptotics, without an exhaustive numerical investigation.

Declaration of Competing Interest

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

Acknowledgments

This work was conducted with support of NASA's Transformative Aeronautics Concepts Program as part of the Convergent Aeronautics Solutions Project and the Transformative Tools and Technology Project.

J.M Emery and J.D. Hochhalter (partially) were supported for this work by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, L.L.C., a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] Henri É Tresca. Mémoire sur l'écoulement des corps solides soumis à de fortes pressions. Gauthier-Villars, 1864. URL https://books.google.com/books?id=4hP9mAEACAAJ.

[2] von Mises Richard. Mechanik der festen körper im plastisch-deformablen zustand. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen. Mathematisch-Physikalische Klasse 1913;582–592:1913. URL http://eudml.org/doc/58894.

[3] Hill Rodney. A theory of the yielding and plastic flow of anisotropic metals. Proc R Soc Lond A 1948;193(1033):281–97.

[4] Haj-Ali Rami, Pecknold David A, Ghaboussi Jamshid, Voyiadjis George Z. Simulated micromechanical models using artificial neural networks. J Eng Mech 2001;127(7):730–8.

[5] Javadi AA, Tan TP, Zhang M. Neural network for constitutive modelling in finite element analysis. Comput Assisted Mech Eng Sci 2003;10(4):523–30.

[6] Jung Sungmoon, Ghaboussi Jamshid. Neural network constitutive model for rate-dependent materials. Comput Struct 2006;84(15–16):955–63.

[7] Kirchdoerfer Trenton, Ortiz Michael. Data-driven computational mechanics. Comput Methods Appl Mech Eng 2016;304:81–101.

[8] Chinesta Francisco, Ladeveze Pierre, Ibanez Ruben. Jose Vicente Aguado, Emmanuelle Abisset-Chavanne, and Elias Cueto. Data-driven computational plasticity. Proc Eng 2017;207:209–14.

[9] Eggersmann Robert, Kirchdoerfer Trenton, Reese Stefanie, Stainier Laurent, Ortiz Michael. Model-free data-driven inelasticity. Comput Methods Appl Mech Eng 2019;350:81–99.

[10] Soare Stefan, Yoon Jeong Whan, Cazacu Oana. On the use of homogeneous polynomials to develop anisotropic yield functions with applications to sheet forming. Int J Plast 2008;24(6):915–44.

[11] Soare Stefan, Barlat Frédéric. Convex polynomial yield functions. J Mech Phys Solids 2010;58(11):1804–18.

[12] Ibáñez Rubén, Abisset-Chavanne Emmanuelle, González David, Duval Jean-Louis, Cueto Elias, Chinesta Francisco. Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. Int J Mater Form 2019;12 (4):717–25.

[13] Raissi Maziar, Karniadakis George Em. Hidden physics models: Machine learning of nonlinear partial differential equations. J Comput Phys 2018;357:125–41. https://doi.org/10.1016/j.jcp.2017.11.039. ISSN 0021-9991.

[14] Raissi Maziar, Perdikaris Paris, Karniadakis George E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.

[15] Samaniego Esteban, Anitescu Cosmin, Goswami Somdatta, Nguyen-Thanh Vien Minh, Guo Hongwei, Hamdia Khader, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput Methods Appl Mech Eng 2020;362:112790.

[16] Hu Jianjun, Goodman Erik D, Seo Kisung. Continuous hierarchical fair competition model for sustainable innovation in genetic programming. In: Genetic Programming Theory and Practice. Springer; 2003. p. 81–98.

[17] Fernández Francisco, Tomassini Marco, Vanneschi Leonardo. An empirical study of multipopulation genetic programming. Genet Program Evolvable Mach 2003;4(1):21–51.

[18] Schmidt Michael D, Lipson Hod. Coevolution of fitness predictors. IEEE Trans Evol Comput 2008;12(6):736–49.

[19] Schmidt Michael, Lipson Hod. Age-fitness pareto optimization. In: Genetic Programming Theory and Practice VIII. Springer; 2011. p. 129–46.

[20] McConaghy Trent. Ffx: Fast, scalable, deterministic symbolic regression technology. In: Genetic Programming Theory and Practice IX. Springer; 2011. p. 235–60.

[21] Schmidt Michael, Lipson Hod. Distilling free-form natural laws from experimental data. Science 2009;324(5923):81–5.

[22] Schoenauer Marc, Sebag Michele, Jouve Francois, Lamy Bertrand, Maitournam Habibou, et al. Evolutionary identification of macro-mechanical models. Adv Genetic Programm 1996;2:467–88.

[23] O'Neill Michael, Vanneschi Leonardo, Gustafson Steven, Banzhaf Wolfgang. Open issues in genetic programming. Genet Program Evolvable Mach 2010;11 (3–4):339–63.

[24] Evett Matthew, Fernandez Thomas. Numeric mutation improves the discovery of numeric constants in genetic programming. Genetic Programm 1998:66–71.

[25] Giustolisi Orazio, Savic Dragan A. A symbolic data-driven technique based on evolutionary polynomial regression. J Hydroinformat 2006;8(3):207–22.

[26] Searson Dominic P, Leahy David E, Willis Mark J. GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 1, Citeseer, 2010. p. 77–80.

[27] Kommenda Michael, Kronberger Gabriel, Winkler Stephan, Affenzeller Michael, Wagner Stefan. Effects of constant optimization by nonlinear least squares minimization in symbolic regression. In: Proceedings of the 15th

annual conference companion on Genetic and evolutionary computation. ACM. p. 1121–8.

[28] Kotha Shravan, Ozturk Deniz, Ghosh Somnath. Parametrically homogenized constitutive models (phcms) from micromechanical crystal plasticity fe simulations, part i: Sensitivity analysis and parameter identification for titanium alloys. Int J Plast 2019;120:296–319.

[29] Ozturk Deniz, Kotha Shravan, Pilchak Adam L, Ghosh Somnath. Two-way multi-scaling for predicting fatigue crack nucleation in titanium alloys using parametrically homogenized constitutive models. J Mech Phys Solids 2019;128:181–207.

[30] Kotha Shravan, Ozturk Deniz, Ghosh Somnath. Parametrically homogenized constitutive models (phcms) from micromechanical crystal plasticity fe simulations: Part ii: Thermo-elasto-plastic model with experimental validation for titanium alloys. Int J Plast 2019;120:320–39.

[31] Versino Daniele, Tonda Alberto, Bronkhorst Curt A. Data driven modeling of plastic deformation. Comput Methods Appl Mech Eng 2017;318:981–1004.

[32] Cannon Andrew H, Hochhalter Jacob D, Mello Alberto W, Bomarito Geoffrey F, Sangid Michael D. Microstamping for improved speckle patterns to enable digital image correlation. Microsc Microanal 2015;21(S3):451–2.

[33] Ruggles Timothy J, Bomarito Geoffrey F, Cannon Andrew H, Hochhalter Jacob D. Selectively electron-transparent microstamping toward concurrent digital image correlation and high-angular resolution electron backscatter diffraction (EBSD) analysis. Microsc Microanal 2017;23(6):1091–5.

[34] Hochhalter Jacob D, Bomarito Geoffrey F, Yeratapally Saikumar R, Leser Patrick E, Ruggles Timothy J, Warner James E, et al. Non-deterministic methods of crystal plasticity-based model calibration. In: S. Ghosh, editor, Integrated Computational Materials Engineering (ICME) - Advancing Computational and Experimental Methods chapter 10, 2020.

[35] Bomarito GF, Warner DH. Micromechanical investigation of ductile failure in Al 5083–H116 via 3D unit cell modeling. J Mech Phys Solids 2015;74:97–110.

[36] Bomarito GF, Warner DH. Predicting the ductile failure of Al5083-H116 specimens with a mechanistic model and no free fitting parameters. Int J Solids Struct 2017;112:25–34.

[37] Wang John T, Bomarito Geoffrey F. Plasticity tool for predicting shear nonlinearity of unidirectional laminates under multiaxial loading. In: Proceedings of the American Society of Composites 31st Technical Conference.

[38] Geers Marc GD, Kouznetsova Varvara G, Brekelmans WAM. Multi-scale computational homogenization: Trends and challenges. J Comput Appl Math 2010;234(7):2175–82.

[39] Schmidt Michael, Lipson Hod. Symbolic regression of implicit equations. In: Genetic Programming Theory and Practice VII. Springer; 2010. p. 73–85.

[40] Hills Daniel JA, Grütter Adrian M, Hudson Jonathan J. An algorithm for discovering Lagrangians automatically from data. PeerJ Computer. Science 2015;1:e31.

[41] Gorry Peter A. General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method. Anal Chem 1990;62(6):570–3.

[42] Koza John R. Genetic programming as a means for programming computers by natural selection. Statist Comput 1994;4(2):87–112.

[43] Worm Tony, Chiu Kenneth. Prioritized grammar enumeration: symbolic regression by dynamic programming. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation. ACM; 2013. p. 1021–8.

[44] Schmidt Michael, Lipson Hod. Comparison of tree and graph encodings as function of problem complexity. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM; 2007. p. 1674–9.

[45] Moré Jorge J. The Levenberg-Marquardt algorithm: implementation and theory. In: Numerical Analysis. Springer; 1978. p. 105–16.

[46] Vinícius Veloso de Melo, Benjamin Fowler, Wolfgang Banzhaf. Evaluating methods for constant optimization of symbolic regression benchmark problems. In: Intelligent Systems (BRACIS), 2015 Brazilian Conference on, IEEE, 2015. p. 25–30.

[47] Fernandezi Francisco, Spezzano Giandomenico, Tomassini Marco, Vanneschi Leonardo. Parallel genetic programming. Parallel Metaheurist: New Class Algorithms 2005;47:127.

[48] Mahfoud Samir W. Niching methods for genetic algorithms. Urbana 1995;51 (95001):62–94.

[49] Martin WN, Lienig Jens, Cohoon James P. Island (migration) models: evolutionary algorithms based on punctuated equilibria. B ack et al. BFM97]. Seiten C 1997;6:101–24.

[50] Piszcz Alan, Soule Terence. Genetic programming: Optimal population sizes for varying complexity problems. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM; 2006. p. 953–4.

[51] Samples Michael E, Daida Jason M, Byom Matt, Pizzimenti Matt. Parameter sweeps for exploring GP parameters. In: Proceedings of the 7th annual workshop on Genetic and evolutionary computation. ACM; 2005. p. 212–9.

[52] Schmidt Michael D, Lipson Hod. Incorporating expert knowledge in evolutionary search: a study of seeding methods. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. p. 1091–8.

[53] Gurson Arthur L. Continuum theory of ductile rupture by void nucleation and growth: Part I-Yield criteria and flow rules for porous ductile media. J Eng Mater Technol 1977;99(1):2–15.

[54] Tvergaard Viggo, Needleman Alan. Analysis of the cup-cone fracture in a round tensile bar. Acta Metall 1984;32(1):157–69.

[55] Bai Yuanli, Wierzbicki Tomasz. A new model of metal plasticity and fracture with pressure and lode dependence. Int J Plasticity 2008;24(6):1071–96.