On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning

Rocha, I. B.C.M.; Kerfriden, P.; van der Meer, F. P.

DOI
10.1016/j.jcpx.2020.100083

Publication date
2021

Document Version
Final published version

Published in
Journal of Computational Physics: X

Journal of Computational Physics: X 9 (2021) 100083

Contents lists available at ScienceDirect

Journal of Computational Physics: X

www.elsevier.com/locate/jcpx

On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning

I.B.C.M. Rocha [a,*], P. Kerfriden [b,c], F.P. van der Meer [a]

[a] Delft University of Technology, Faculty of Civil Engineering and Geosciences, P.O. Box 5048, 2600GA Delft, the Netherlands
[b] Mines ParisTech (PSL University), Centre des matériaux, 63-65 Rue Henri-Auguste Desbruères BP87, F-91003 Évry, France
[c] Cardiff University, School of Engineering, Queen's Buildings, The Parade, Cardiff, CF24 3AA, United Kingdom

ARTICLE INFO

Article history:
Received 10 June 2020
Received in revised form 5 October 2020
Accepted 21 December 2020
Available online 28 December 2020

Keywords:
Concurrent multiscale
Surrogate modeling
Probabilistic learning
Gaussian Processes (GP)
Active learning

ABSTRACT

Concurrent multiscale finite element analysis (FE^2) is a powerful approach for high-fidelity modeling of materials for which a suitable macroscopic constitutive model is not available. However, the extreme computational effort associated with computing a nested micromodel at every macroscopic integration point makes FE^2 prohibitive for most practical applications. Constructing surrogate models able to efficiently compute the microscopic constitutive response is therefore a promising approach in enabling concurrent multiscale modeling. This work presents a reduction framework for adaptively constructing surrogate models for FE^2 based on statistical learning. The nested micromodels are replaced by a machine learning surrogate model based on Gaussian Processes (GP). The need for *offline* data collection is bypassed by training the GP models *online* based on data coming from a small set of fully-solved *anchor* micromodels that undergo the same strain history as their associated macroscopic integration points. The Bayesian formalism inherent to GP models provides a natural tool for *online* uncertainty estimation through which new observations or inclusion of new *anchor* micromodels are triggered. The surrogate constitutive manifold is constructed with as few micromechanical evaluations as possible by enhancing the GP models with gradient information and the solution scheme is made robust through a greedy data selection approach embedded within the conventional finite element solution loop for nonlinear analysis. The sensitivity to model parameters is studied with a tapered bar example with plasticity and the framework is further demonstrated with the elastoplastic analysis of a plate with multiple cutouts and with a crack growth example for mixed-mode bending. Although not able to handle non-monotonic strain paths in its current form, the framework is found to be a promising approach in reducing the computational cost of FE^2, with significant efficiency gains being obtained without resorting to *offline* training.

* Corresponding author.
E-mail address: i.rocha@tudelft.nl (I.B.C.M. Rocha).

https://doi.org/10.1016/j.jcpx.2020.100083

1. Introduction

There is a growing demand for high-fidelity numerical techniques capable of describing material behavior across spatial scales. With recent advances in additive manufacturing allowing for the development of novel materials with highly-tailored microstructures [1], multiscale modeling techniques will become increasingly relevant in the design of novel materials and structures. One popular approach for concurrent multiscale modeling is the so-called FE^2 approach [2,3], in which macroscopic material response is directly upscaled from embedded micromodels without introducing additional constitutive assumptions. FE^2 is a powerful and versatile technique used in a number of solid mechanics applications, from continuous [4] and discontinuous [5] mechanical equilibrium to multiphysics problems involving heat and mass transfer [6] and material degradation due to aging [7,8]. However, FE^2 has the major drawback of being associated with extreme computational costs, hindering its application in actual design scenarios. Enabling the use of FE^2 in many practical applications that would benefit from its accuracy and versatility is therefore highly contingent on being able to reduce its computational cost to tractable levels.

A promising approach in accelerating FE^2 models consists in constructing surrogate models that take the place of the original high-fidelity micromodels at each macroscopic integration point. When building surrogates, the goal is to maintain as much of the generality offered by the original micromodel while eliminating as much computational complexity as possible. One option is to employ unsupervised learning on a number of full-order solution snapshots in order to define lower-dimensional solution manifolds for both displacements [9,10] and internal forces [11,12] at the microscale [13–15]. Alternatively, a supervised learning approach can be taken by using snapshots of the homogenized micromodel response to directly define a data-driven regression model for the macroscopic constitutive behavior [16–21]. Although resulting in models of distinct natures, both approaches rely on the existence of an observation database on the behavior of the original micromodel that is usually obtained *offline* (before deployment on a multiscale setting) and should cover every possible scenario the surrogate is expected to approximate *online*.

However, building such a database of model snapshots can be a challenging task (see [20,22] for interesting approaches based on Design of Experiments and [14] for a data-driven training framework based on Bayesian Optimization). For micromodels employing path-dependent materials, this *offline training* process entails sampling a highly-complex constitutive manifold that depends on an arbitrarily long strain history and can therefore be excessively sensitive to small changes in boundary conditions (*e.g.* strain localization and crack propagation problems). Furthermore, training a surrogate with such a complex dataset often requires additional partitioning techniques in order to avoid computationally inefficient reduced models [23,24]. An alternative to the conventional *offline-online* approach that has been gaining popularity is the use of adaptive reduction frameworks that either preclude the need for *offline* training altogether [25,26] or combine different reduction techniques into a single framework with only limited *offline* effort while employing *online* error indicators to continuously assess the quality of the approximation and trigger a refinement of the surrogates when necessary. Nevertheless, obtaining consistent adaptivity criteria for either hyper-reduced models [25] or machine learning models based on least-squares solutions [19,27] is not straightforward. At the other end of the spectrum, works dealing with constitutive models based on Bayesian regression techniques, that provide a natural way to estimate error [28–30], do not often take advantage of their potential for creating adaptive frameworks (see [31,32] for an interesting framework specifically tailored to crystal plasticity). There is a need, therefore, for the development of fully-online approaches for general FE^2 models with reliable adaptivity strategies based on sequential learning techniques and on error estimation methods with robust probabilistic foundations.

This work presents an adaptive probabilistic framework for constructing surrogate constitutive models for nonlinear concurrent multiscale analysis. The approach is based on substituting the original models associated to macroscopic integration points with a machine learning surrogate. In order to bypass the need for an *offline* training phase, a small number of fully-solved models associated to representative macroscopic integration points are used to generate constitutive data *online*. Because these models are not directly used to make predictions at every macroscopic iteration but only provide an indirect coupling between the scales, we denote them as *anchor models* (Fig. 1). Stress and stiffness data resulting from subjecting the *anchor* models to the same strain histories seen by their respective anchoring points is used to build a surrogate model based on the Gaussian Process (GP) regression technique. The framework preserves the generality of FE^2 by not assuming the existence of any internal variables at the macroscale and builds surrogates that take only strains as input. However, this implies a one-to-one mapping between stresses and strains, making the framework in its current form unsuitable for treating non-monotonic load paths. This can be addressed in future extensions of the framework by employing autoregressive GPs [33,34].

The accuracy of this reduced-order solution is controlled with uncertainty information that arises naturally from the Bayesian formalism of the GP models. The resultant material model is embedded within a conventional finite element solution in a way that ensures the macroscopic solution is numerically robust and limits the sampling of new data as much as possible in order to maximize efficiency. The framework is demonstrated with a set of numerical tests including both bulk strain localization and crack propagation in order to assess its accuracy, efficiency and versatility.

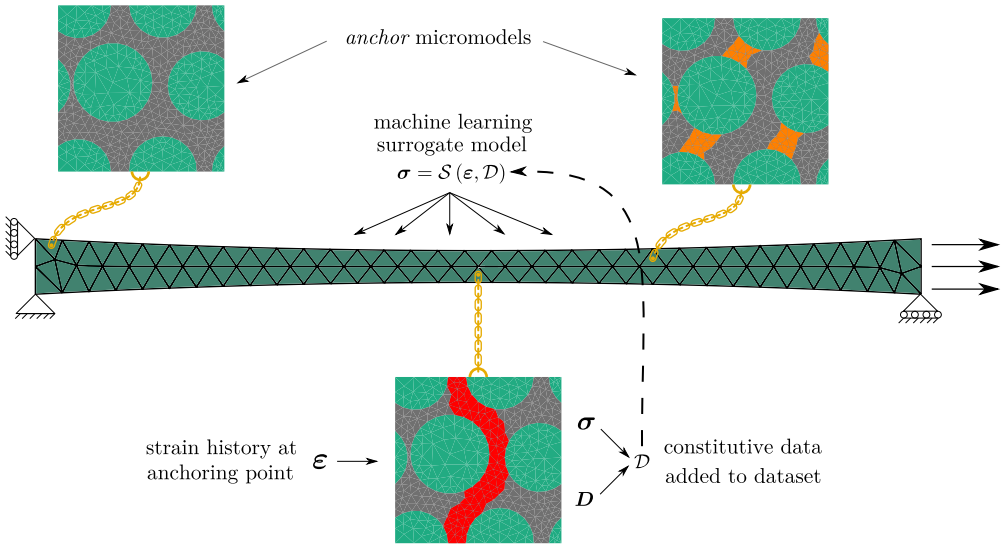


Fig. 1. Schematic representation of the *online* adaptive reduction framework presented in this work. A small number of *anchor* models — in this case a matrix reinforced with circular inclusions — are used to train a machine learning surrogate that evolves as the macroscopic structure is loaded. Adaptive sampling guarantees that anchor models operating at novel regions in parameter space — illustrated here by plastic strain bands shown in orange and red — will populate the dataset $\mathcal{D}$ more often.

2. Concurrent multiscale analysis

2.1. Macroscopic problem

We begin by briefly introducing the concurrent multiscale equilibrium problem that we seek to accelerate. Let Ω define the macroscopic domain being modeled. We wish to find the displacement field $\mathbf{u}^{\Omega}$ resulting from a set of Dirichlet and Neumann boundary conditions applied to the surface Γ that bounds Ω. Under the assumption of small strains, the equilibrium solution is found by satisfying:

$$\operatorname{div}\left(\boldsymbol{\sigma}^{\Omega}\right) = \mathbf{0} \qquad \boldsymbol{\varepsilon}^{\Omega} = \frac{1}{2}\left(\nabla\mathbf{u}^{\Omega} + \left(\nabla\mathbf{u}^{\Omega}\right)^{\mathrm{T}}\right) \tag{1}$$

where $\operatorname{div}(\cdot)$ is the divergence operator, $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are respectively the stress and strain tensors and body forces have been neglected. In order to solve for $\mathbf{u}^{\Omega}$, a constitutive model $\mathcal{M}$ that relates $\boldsymbol{\sigma}^{\Omega}$ and $\boldsymbol{\varepsilon}^{\Omega}$ must be introduced:

$$\boldsymbol{\sigma}^{\Omega} = \mathcal{M}\left(\boldsymbol{\varepsilon}^{\Omega}, \boldsymbol{\varepsilon}_{\mathrm{h}}^{\Omega}\right) \tag{2}$$

where $\boldsymbol{\varepsilon}_{\mathrm{h}}^{\Omega}$ is a history term that accounts for strain path dependency. In the context of multiscale analysis, the model $\mathcal{M}$ can be seen as a homogenization operator that lumps all physical processes happening at scales lower than Ω into a homogeneous medium with equivalent behavior. Depending on how complex the microscopic behavior is, $\mathcal{M}$ can range from having a relatively simple form (*e.g.* linear elasticity) to being next to impossible to formulate explicitly.

2.2. Microscopic problem

In a concurrent multiscale approach, we do not formulate $\mathcal{M}$ directly and instead opt for upscaling microscopic behavior from micromodels embedded at each macroscopic material point. Let ω be a Representative Volume Element (RVE) of the microscopic material features whose behavior we would like to upscale. Assuming the principle of separation of scales is valid ($\omega \ll \Omega$), we can link the two scales by enforcing:

$$\mathbf{u}^{\omega} = \boldsymbol{\varepsilon}^{\Omega}\mathbf{x}^{\omega} + \widetilde{\mathbf{u}} \tag{3}$$

where the response is decomposed into a linear displacement field imposed by the presence of macroscopic strains ($\mathbf{x}^{\omega}$ are the microscopic spatial coordinates) and a fluctuation field $\widetilde{\mathbf{u}}$ that accounts for microscopic inhomogeneities.

In order to obtain $\boldsymbol{\sigma}^{\Omega}$, we first find an equilibrium solution for $\mathbf{u}^{\omega}$ by satisfying:

$$\operatorname{div}\left(\boldsymbol{\sigma}^{\omega}\right) = \mathbf{0} \qquad \boldsymbol{\varepsilon}^{\omega} = \frac{1}{2}\left(\nabla\mathbf{u}^{\omega} + \left(\nabla\mathbf{u}^{\omega}\right)^{\mathrm{T}}\right) \tag{4}$$

where we see that once more the need arises for the definition of a constitutive model, this time relating $\boldsymbol{\varepsilon}^{\omega}$ with $\boldsymbol{\sigma}^{\omega}$. Underlying the choice for a multiscale approach is the assumption that constitutive behavior can be represented by models of decreasing complexity as one descends to lower scales. It is therefore common to employ regular constitutive models (*e.g.* (visco)elasticity, (visco)plasticity, damage) for material components at the microscale [4,6–8]. However, the framework is flexible in allowing for models on an even lower third scale to be embedded to material points in ω (at the cost of even higher computational effort).

2.3. Bulk homogenization

With a solution for $\mathbf{u}^{\omega}$ under the constraint that the fluctuation field $\widetilde{\mathbf{u}}$ must be periodic, we can use the Hill-Mandel principle [35] to obtain homogenization expressions for the macroscopic stresses $\boldsymbol{\sigma}^{\Omega}$ and consistent tangent stiffness $\mathbf{D}^{\Omega}$:

$$\boldsymbol{\sigma}^{\Omega} = \frac{1}{\omega}\int_{\omega} \boldsymbol{\sigma}^{\omega} \mathrm{d}\omega \qquad \mathbf{D}^{\Omega} = \mathcal{P}\left(\mathbf{K}^{\omega}\right) \tag{5}$$

from which we obtain the intuitive result that the macroscopic stresses are simply the volume average of the microscopic ones. Finally, the macroscopic constitutive tangent stiffness is computed through a probing operator $\mathcal{P}$ applied on the global microscopic tangent stiffness matrix $\mathbf{K}^{\omega}$ [5].

2.4. Cohesive homogenization

Although the preceding formulation allows for general microscopic constitutive behavior to be upscaled, the response loses objectivity with respect to the RVE size after the onset of global microscopic softening [36], *i.e.* when the determinant of the acoustic tensor is zero along a given direction $\mathbf{n}$:

$$\det\left(\mathbf{n}^{\mathrm{T}}\mathbf{D}^{\Omega}\mathbf{n}\right) = 0 \tag{6}$$

This non-objectivity arises because the volume ω_{d} of the strain localization band that causes the softening does not scale with the RVE size, an observation that motivates the use of a modified version of the Hill-Mandel principle [5,36]:

$$\frac{1}{w}\boldsymbol{\tau}^{\Omega}\delta[\![\mathbf{v}]\!]^{\Omega} = \frac{1}{wh}\int_{\omega_{\mathrm{d}}} \boldsymbol{\sigma}^{\omega}\delta\boldsymbol{\varepsilon}^{\omega}\mathrm{d}\omega \tag{7}$$

where w and h are geometric RVE parameters that depend on the localization band orientation, $\boldsymbol{\tau}^{\Omega}$ is a macroscopic traction and $[\![\mathbf{v}]\!]^{\Omega}$ is a shifted displacement jump that allows for an initially-rigid cohesive response. Note that the homogenization is now performed towards a cohesive traction acting on a macroscopic surface that defines a discontinuity in $\mathbf{u}^{\Omega}$.

The development of a consistent strategy for continuous-discontinuous scale linking in FE^2 is an open issue that is left out of the scope of the present discussion, with a number of different approaches being found, for instance, in [5,36–39]. For the purpose of building surrogate models for the RVE response, it suffices to acknowledge that two distinct models should be trained for bulk and cohesive responses, as the underlying constitutive manifolds have dimensions with different physical interpretations (strain/stress versus jump/traction).

2.5. Acceleration strategy

Assuming FEM is used to solve the mechanical problems at both scales, we can approximate the computational cost of a single macroscopic iteration as:

$$\mathrm{Cost} \approx \left(N_{\mathrm{dof}}^{\Omega}\right)^{x} + N_{\mathrm{ip}}^{\Omega} N_{\mathrm{iter}}^{\omega}\left(N_{\mathrm{dof}}^{\omega}\right)^{x} \tag{8}$$

where N_{dof} represents the number of degrees of freedom of the model, N_{ip}^{Ω} is the number of macroscopic integration points and $N_{\mathrm{iter}}^{\omega}$ is the number of iterations necessary for convergence of the microscopic BVP. We assume for simplicity that the bulk of the effort comes from solving the linearized systems of equations involving $\mathbf{K}^{\Omega}$ and $\mathbf{K}^{\omega}$ for nodal displacements, where the complexity exponent x depends on the solver used. It can be seen that the second term, associated with solving the microscopic equilibrium problems, quickly outweighs the first and becomes a performance bottleneck as $N_{\mathrm{dof}}^{\omega}$ increases, especially since the number of macroscopic integration points N_{ip}^{Ω} increases together with $N_{\mathrm{dof}}^{\Omega}$. Constructing a surrogate that replaces the original constitutive model $\mathcal{M}$ of Eq. (2) is therefore an effective approach to accelerating FE^2.

However, constructing such a surrogate *offline* is a challenging undertaking, otherwise there would not have been need for a multiscale approach in the first place. From Eq. (2) we see that the constitutive manifold to be reproduced can have an arbitrarily high dimensionality due to the dependency on $\boldsymbol{\varepsilon}_{\mathrm{h}}^{\Omega}$. This is equivalent to stating that the shape of the $\boldsymbol{\varepsilon}^{\Omega}$-$\boldsymbol{\sigma}^{\Omega}$ manifold can change after each load step. Sampling this high-dimensional input space *offline* in order to have a surrogate

that is accurate for arbitrary strain histories seems to be an intractable problem that, to the best of our knowledge, has still not been tackled in a satisfactory way.

This issue can be avoided by exploiting the fact that $\boldsymbol{\varepsilon}^{\Omega}$ (and therefore $\boldsymbol{\varepsilon}_{\mathrm{n}}^{\Omega}$) are often highly constrained by the geometry and boundary conditions of the macroscopic structure being modeled [24]. We therefore opt for constructing a highly-tailored surrogate model $\mathcal{S}$ *online* based on a dataset $\mathcal{D}$ of observations coming from a small number of fully-solved micromodels:

$$\boldsymbol{\sigma}^{\Omega} = \mathcal{S}\left(\boldsymbol{\varepsilon}^{\Omega}, \mathcal{D}\right) \tag{9}$$

which can then be used to compute the constitutive response for a fraction of the cost. When trying to keep $\mathcal{D}$ as small as possible for the case at hand, it is crucial to have a means for quantifying the uncertainty in probing $\mathcal{S}$ for any given $\boldsymbol{\varepsilon}^{\Omega}$. In this work, a Bayesian approach is adopted to assess *online* whether $\mathcal{D}$ is large enough to provide the desired level of confidence in $\mathcal{S}$ at a given $\boldsymbol{\varepsilon}^{\Omega}$.

3. Bayesian surrogate modeling

In this section we introduce the Bayesian regression approach used to construct surrogate constitutive models, beginning from parametric versions of the surrogate model $\mathcal{S}$ — *i.e.* by encapsulating the constitutive information in $\mathcal{D}$ into a set of parameters $\mathbf{w}$ — and eventually moving to a non-parametric model based on Gaussian Processes (GP) that uses the data in $\mathcal{D}$ directly in order to make predictions. Our goal here is to appeal to the reader who might be unfamiliar with probabilistic regression models by starting from classical least-squares regression and gradually moving towards a Bayesian approach. Nevertheless, the discussion is kept as brief and focused as possible. The interested reader can find richer discussions on the subject in [40,41].

3.1. Least-squares regression

We start by building a parametric model $y(\mathbf{w}, \mathbf{x})$ that approximates a scalar target response t by fitting $\mathbf{w}$ with a dataset $\mathcal{D}$ of N observations $\mathbf{t}_{\mathrm{o}}$ at $\mathbf{X}_{\mathrm{o}} = [\mathbf{x}_{\mathrm{o}1} \cdots \mathbf{x}_{\mathrm{o}N}]$. We therefore assume that the target t can be written as:

$$t = y(\mathbf{w}, \mathbf{x}) + \epsilon \quad \text{with} \quad y(\mathbf{w}, \mathbf{x}) = \sum_{j}^{M} w_j \phi_j(\mathbf{x}) \text{ and } p(\epsilon) = \mathcal{N}\left(\epsilon | 0, \sigma_{\mathrm{n}}^2\right) \tag{10}$$

where the noise ϵ is given by a zero-mean Gaussian distribution with variance σ_{n}^2 and y is linear with respect to its M weights, gathered in the vector $\mathbf{w}$. Note that the assumption of a linear model does not limit its fitting capabilities since the basis functions $\boldsymbol{\phi}$ define a feature space that can be nonlinear in $\mathbf{x}$. In the context of this work, t can be a single stress or traction component, but the discussion is equally applicable to the problem of modeling individual model components (*e.g.* yield parameters).

In order to find values for $\mathbf{w}$, we compute the *likelihood function* of the model, *i.e.* how likely the model is to produce the values $\mathbf{t}_{\mathrm{o}}$ in $\mathcal{D}$ given $\mathbf{w}$:

$$p\left(\mathbf{t}_{\mathrm{o}} | \mathbf{w}, \sigma_{\mathrm{n}}^2\right) = \prod_{i}^{N} \mathcal{N}\left(t_{\mathrm{o}i} | \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{\mathrm{o}i}), \sigma_{\mathrm{n}}^2\right) \tag{11}$$

which is a product of the probabilities of each point in isolation because we assume that each sample $t_{\mathrm{o}i}$ is sampled from the conditional distribution $p(t|y)$ independently. We can find an optimum data fit for $\mathbf{w}$ by maximizing the likelihood and assuming that the shapes of $\boldsymbol{\phi}$ are fixed *a priori*:

$$\nabla_{\mathbf{w}} p\left(\mathbf{t}_{\mathrm{o}} | \mathbf{w}, \sigma_{\mathrm{n}}^2\right) = 0 \quad \Rightarrow \quad \mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}_{\mathrm{o}} \text{ and } \sigma_{\mathrm{n}}^2 = \frac{1}{N} \sum_{i}^{N} \left(t_{\mathrm{o}i} - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{\mathrm{o}i})\right)^2 \tag{12}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$ is a matrix with basis function values $\phi_{i,j} = \phi_i\left(\mathbf{x}_{\mathrm{o}j}\right)$ evaluated at each point in $\mathcal{D}$. Note that this is equivalent to minimizing the sum of squared differences between y and t, so $\mathbf{w}_{\mathrm{ML}}$ is the same parameter vector obtained by the classical least squares approach and σ_{n}^2 quantifies the spread of t around y.

Here we do not opt for a least-squares approach for three reasons. Firstly, it can suffer from severe overfitting when the dataset $\mathcal{D}$ is small — which is the case in the present work since we have no *offline* training and only a small number of fully-solved micromodels to sample from. Secondly, the uncertainty associated with σ_{n}^2 is constant throughout the input space $\mathbf{x}$ and does not provide an indication that the model is being used at a location far from data points (which could then be used to trigger a refinement of $\mathcal{D}$). Finally, performing model selection in the least-squares framework, using cross-validation for instance, is rather tedious compared to performing model selection in the context of Bayesian regression methods [40].

3.2. Bayesian parametric regression

In the Bayesian approach to regression, we not only assume an uncertainty over the target t but also over the weights $\mathbf{w}$. We initially assume a prior probability over $\mathbf{w}$ that represents our initial model assumptions before any data is encountered:

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}|\mathbf{0}, \sigma_{\mathrm{w}}^2\mathbf{I}\right) \tag{13}$$

where σ_{w}^2 is the variance parameter associated with the uncertainty over values of $\mathbf{w}$. Information from $\mathcal{D}$ is incorporated by using Bayes' theorem to obtain a posterior probability distribution for $\mathbf{w}$:

$$p(\mathbf{w}|\mathbf{t}_{\mathrm{o}}) = \frac{p(\mathbf{t}_{\mathrm{o}}|\mathbf{w})\, p(\mathbf{w})}{p(\mathbf{t}_{\mathrm{o}})} \quad \text{with} \quad p(\mathbf{t}_{\mathrm{o}}) = \int p(\mathbf{t}_{\mathrm{o}}|\mathbf{w})\, p(\mathbf{w})\, \mathrm{d}\mathbf{w} \tag{14}$$

where $p(\mathbf{t}_{\mathrm{o}}|\mathbf{w})$ is the likelihood function of Eq. (11) and $p(\mathbf{t}_{\mathrm{o}})$ is the *marginal likelihood* of the model (*i.e.* the probability of producing the dataset $\mathcal{D}$). Because both $\mathbf{w}$ and $t|\mathbf{w}$ are Gaussian variables, an analytical solution exists for $p(\mathbf{w}|\mathbf{t}_{\mathrm{o}})$ [40]:

$$p(\mathbf{w}|\mathbf{t}_{\mathrm{o}}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_{\mathrm{N}}, \mathbf{S}_{\mathrm{N}}) \quad \text{with} \quad \mathbf{w}_{\mathrm{N}} = \frac{1}{\sigma_{\mathrm{n}}^2}\mathbf{S}_{\mathrm{N}}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}_{\mathrm{o}} \text{ and } \mathbf{S}_{\mathrm{N}} = \left(\frac{1}{\sigma_{\mathrm{w}}^2}\mathbf{I} + \frac{1}{\sigma_{\mathrm{n}}^2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1} \tag{15}$$

Note that the expected value of $\mathbf{w}$ now depends on both the data points in $\mathcal{D}$ and on our initial beliefs about $\mathbf{w}$ represented by the prior distribution. Given this posterior, the best guess for $\mathbf{w}$ is the one with the highest $p(\mathbf{w}|\mathbf{t}_{\mathrm{o}})$. This is the so-called Maximum A Posteriori (MAP) value and, for a Gaussian posterior, $\mathbf{w}_{\mathrm{MAP}} = \mathbf{w}_{\mathrm{N}}$. This estimation for $\mathbf{w}$ is equivalent to a least-squares prediction with a quadratic regularization term proportional to σ_{w}^2 which helps reducing the negative effects of overfitting.

However, in a fully Bayesian treatment of linear regression, we do not choose one specific value for $\mathbf{w}$ but rather make a prediction for a new target value t_* by averaging over all possible values of $\mathbf{w}$:

$$p\left(t_*|\mathbf{t}_{\mathrm{o}}, \sigma_{\mathrm{n}}^2, \sigma_{\mathrm{w}}^2\right) = \int p\left(t_*|\mathbf{w}, \sigma_{\mathrm{n}}^2\right) p\left(\mathbf{w}|\mathbf{t}_{\mathrm{o}}, \sigma_{\mathrm{w}}^2, \sigma_{\mathrm{n}}^2\right) \mathrm{d}\mathbf{w} \tag{16}$$

In order to set the stage for Gaussian Processes, it is also interesting to represent the expectation of t_* as:

$$\mathbb{E}[t_*|\mathbf{x}_*] = \sum_i^N k(\mathbf{x}_{\mathrm{o}i}, \mathbf{x}_*)\, t_{\mathrm{o}i} \quad \text{with} \quad k(\mathbf{x}_p, \mathbf{x}_q) = \frac{1}{\sigma_{\mathrm{n}}^2}\boldsymbol{\phi}^{\mathrm{T}}(\mathbf{x}_p)\,\mathbf{S}_{\mathrm{N}}\boldsymbol{\phi}(\mathbf{x}_q) \tag{17}$$

where $\mathbf{w}$ has now vanished and the new prediction is cast as a linear combination of values from $\mathcal{D}$, where $k(\mathbf{x}_p, \mathbf{x}_q)$ is a *kernel* that measures the similarity between two points in input space.

The Bayesian approach to the (generalized) linear model circumvents the problem of overfitting and provides confidence intervals. However, the linear model has the drawback of having a fixed number of basis functions $\boldsymbol{\phi}$ whose shapes need to be chosen before $\mathcal{D}$ is observed. This can be circumvented by either allowing $\boldsymbol{\phi}$ to change in shape during training (*e.g.* in a neural network, where $\boldsymbol{\phi}$ are neuron values coming from the last hidden layer) or by explicitly choosing a kernel $k(\mathbf{x}_p, \mathbf{x}_q)$ and adopting a distribution over functions instead of over weights, as in Gaussian Process regression models.

3.3. Gaussian Process (GP) regression

Since the regression function y is a function of $\mathbf{w}$, adopting a prior distribution over $\mathbf{w}$ (Eq. (13)) implicitly leads to a distribution over function values $\mathbf{y}$. This is the basis of Gaussian Process (GP) models. Here we assume these function values are jointly Gaussian:

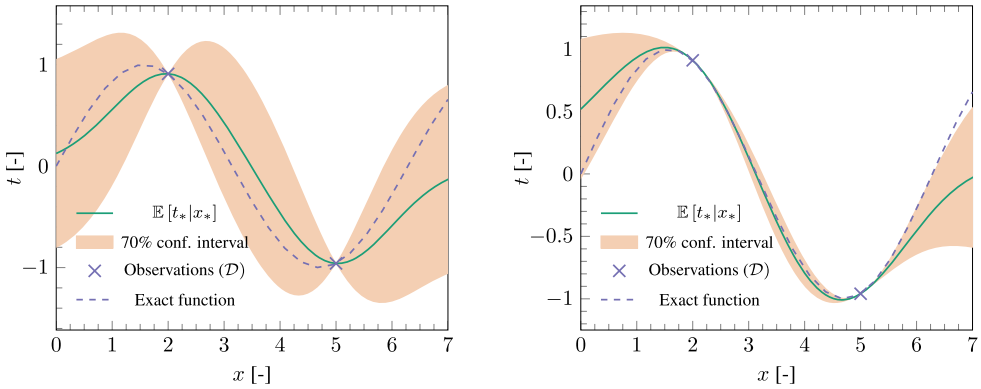
$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, K(\mathbf{X}, \mathbf{X})) \tag{18}$$

where a zero mean is assumed without loss of generality and $K(\mathbf{X}, \mathbf{X})$ is the so-called *Gram* matrix that defines the covariance between the input values $\mathbf{X}$ associated with $\mathbf{y}$ through a kernel $k(\mathbf{x}_p, \mathbf{x}_q)$:

$$K_{pq} = k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_{\mathrm{f}}^2 \exp\left(-\frac{1}{2\ell^2}\left\|\mathbf{x}_p - \mathbf{x}_q\right\|^2\right) \tag{19}$$

where instead of first defining a prior over weights and deriving an equivalent kernel as in Eq. (17) we directly assume a kernel — in this case the *squared exponential* kernel [41] defined by a variance σ_{f}^2 and a length scale ℓ — which determines the level of correlation between points in input space. It can be shown that adopting the squared exponential kernel is equivalent to formulating a parametric model with an infinite number of basis functions $\boldsymbol{\phi}$ or a Bayesian neural network with one hidden layer with an infinite number of neurons [40,41].

With this definition for the covariance structure between function values, we can obtain the joint distribution of training targets $\mathbf{t}_{\mathrm{o}}$ by marginalizing (averaging) over all possible values of $\mathbf{y}_{\mathrm{o}}$:



(a) Only target observations (b) Target and derivative observations

Fig. 2. GP predictions for the function $t = \sin(x)$ constructed with two observations. Inclusion of derivative observations improves predictions around training points and reduces uncertainty.

$$p(\mathbf{t}_o) = \int p(\mathbf{t}_o|\mathbf{y}_o)\, p(\mathbf{y}_o)\, \mathrm{d}\mathbf{y} = \mathcal{N}\left(\mathbf{t}_o|\mathbf{0}, K(\mathbf{X}_o, \mathbf{X}_o) + \sigma_n^2\mathbf{I}\right) \tag{20}$$

which allows us to incorporate information from $\mathcal{D}$ by defining a joint distribution between training values and new predictions for the function value:

$$p\left(\begin{bmatrix}\mathbf{t}_o\\ \mathbf{y}_*\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{t}_o\\ \mathbf{y}_*\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix} K(\mathbf{X}_o, \mathbf{X}_o) + \sigma_n^2\mathbf{I} & K(\mathbf{X}_o, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}_o) & K(\mathbf{X}_*, \mathbf{X}_*)\end{bmatrix}\right) \tag{21}$$

and subsequently find the conditional probability $p(\mathbf{y}_*|\mathbf{t}_o)$ of the new function values given that values coming from $\mathcal{D}$ are known:

$$p(\mathbf{y}_*|\mathbf{t}_o) = \mathcal{N}(\mathbf{y}_*|\mathbf{m}_*, \mathbf{S}_*) \tag{22}$$

which is again a Gaussian distribution with mean and covariance given by [40,41]:

$$\mathbf{m}_* = K(\mathbf{X}_*, \mathbf{X}_o)\left(K(\mathbf{X}_o, \mathbf{X}_o) + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{t}_o \tag{23}$$

$$\mathbf{S}_* = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X}_o)\left(K(\mathbf{X}_o, \mathbf{X}_o) + \sigma_n^2\mathbf{I}\right)^{-1} K(\mathbf{X}_o, \mathbf{X}_*) \tag{24}$$

Defining $\mathbf{k}_* = K(\mathbf{X}_o, \mathbf{x}_*) \in \mathbb{R}^{N\times 1}$ and $\mathbf{K}_o = K(\mathbf{X}_o, \mathbf{X}_o) \in \mathbb{R}^{N\times N}$, we can arrive at shorter expressions for the expectation and variance of the function value at a single new point $\mathbf{x}_*$:

$$\mathbb{E}[y_*|\mathbf{x}_*] = \mathbf{k}_*^{\mathrm{T}}\left(\mathbf{K}_o + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{t}_o \tag{25}$$

$$\mathbb{V}[y_*|\mathbf{x}_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathrm{T}}\left(\mathbf{K}_o + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}_* \tag{26}$$

Note that here we opt for directly using the function value y_* instead of its noisy version t_* to define our surrogate models. This effectively makes the adaptive components of the acceleration framework, which are driven by increases in the variance given by Eq. (26), less sensitive to changes in σ_n^2.

In Fig. 2a we demonstrate a GP model by plotting predictions based on two observations of the noiseless target function $t = \sin(x)$. The conditioning of Eq. (22) guarantees that y_* coincides with the observations at the training points $\mathbf{X}_o$. Away from the training space $\mathcal{D}$ the GP returns to its zero-mean prior with high variance. This behavior is desirable for our application, since this uncertainty can be used as a trigger to either making new observations at the existing full-order integration points or adding new points to the full-order set when necessary.

3.4. Predicting derivatives and including derivative observations

In building constitutive model surrogates, in addition to predicting new function values (Eq. (25)), their derivatives with respect to the input $\mathbf{x}_*$ are also needed. These can be computed by differentiating Eq. (25):

$$\frac{\partial}{\partial \mathbf{x}_*}\mathbb{E}[y_*|\mathbf{x}_*] = \sum_i^N a_i \frac{\partial}{\partial \mathbf{x}_*} k(\mathbf{x}_*, \mathbf{x}_i) \tag{27}$$

where a_i is the i-th component of the vector $\mathbf{a} = \left(\mathbf{K} + \sigma_\mathrm{n}^2\mathbf{I}\right)^{-1}\mathbf{t}$ and the derivative of the kernel is given by:

$$k'\left(\mathbf{x}_p, \mathbf{x}_q\right) \equiv \frac{\partial}{\partial \mathbf{x}_p} k\left(\mathbf{x}_p, \mathbf{x}_q\right) = -\frac{1}{\ell^2}\left(\mathbf{x}_p - \mathbf{x}_q\right) k\left(\mathbf{x}_p, \mathbf{x}_q\right) \tag{28}$$

However, since we can also observe these derivatives (in the form of $\mathbf{D}^\Omega$ from Eq. (5)) at the anchor models, it is also interesting to include this information in the GP in order to improve its predictions. This is achieved by recognizing that the derivative of a GP remains Gaussian and defining the covariance between function values and derivatives as [42]:

$$\mathrm{cov}\left(\mathbf{y}'_p, y_q\right) = \frac{\partial}{\partial \mathbf{x}_p} k\left(\mathbf{x}_p, \mathbf{x}_q\right) \qquad \mathrm{cov}\left(\mathbf{y}'_p, \mathbf{y}'_q\right) = \frac{\partial^2}{\partial \mathbf{x}_q \partial \mathbf{x}_p} k\left(\mathbf{x}_p, \mathbf{x}_q\right) + \sigma_\mathrm{d}^2 \tag{29}$$

where σ_d^2 is a noise parameter that represents the uncertainty associated to observing the derivatives, the first-order derivative of the kernel is given in Eq. (28) and its second derivative is a matrix given by:

$$k''\left(\mathbf{x}_p, \mathbf{x}_q\right) \equiv \frac{\partial^2}{\partial \mathbf{x}_q \mathbf{x}_p} k\left(\mathbf{x}_p, \mathbf{x}_q\right) = \frac{1}{\ell^2}\left(\mathbf{I} - \frac{1}{\ell^2}\left(\mathbf{x}_p - \mathbf{x}_q\right)\left(\mathbf{x}_p - \mathbf{x}_q\right)^\mathrm{T}\right) k\left(\mathbf{x}_p, \mathbf{x}_q\right) \tag{30}$$

With these definitions, the joint prior distribution of training targets becomes [43]:

$$p\left(\begin{bmatrix}\mathbf{t}_\mathrm{o}\\ \mathbf{t}'_\mathrm{o}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{t}_\mathrm{o}\\ \mathbf{t}'_\mathrm{o}\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_\mathrm{o} + \sigma_\mathrm{n}^2\mathbf{I} & \mathbf{K}_\mathrm{td}\\ \mathbf{K}_\mathrm{td}^\mathrm{T} & \mathbf{K}_\mathrm{dd}\end{bmatrix}\right) \equiv \mathcal{N}\left(\begin{bmatrix}\mathbf{t}_\mathrm{o}\\ \mathbf{t}'_\mathrm{o}\end{bmatrix}\middle|\mathbf{0}, \overline{\mathbf{K}}_\mathrm{o}\right) \tag{31}$$

where $\mathbf{K}_\mathrm{o}$ is the same kernel matrix appearing in Eq. (25) and $\mathbf{K}_\mathrm{td} \in \mathbb{R}^{N\times ND}$ and $\mathbf{K}_\mathrm{dd} \in \mathbb{R}^{ND\times ND}$ are composed of blocks of $k'\left(\mathbf{x}_p, \mathbf{x}_q\right)$ and $k''\left(\mathbf{x}_p, \mathbf{x}_q\right)$, respectively, with $\overline{\mathbf{K}}_\mathrm{o}$ being the resultant $N(D+1)\times N(D+1)$ covariance matrix (D is the dimensionality of $\mathbf{x}$). Making point predictions is done in a similar way as in Eq. (25):

$$\overline{\mathbf{k}}_* = \begin{bmatrix}\mathbf{k}_* & k'\left(\mathbf{x}_1, \mathbf{x}_*\right) & \cdots & k'\left(\mathbf{x}_N, \mathbf{x}_*\right)\end{bmatrix}^\mathrm{T} \tag{32}$$

$$\mathbb{E}\left[y_*|\mathbf{x}_*\right] = \overline{\mathbf{k}}_*^\mathrm{T}\overline{\mathbf{K}}_\mathrm{o}^{-1}\overline{\mathbf{t}}_\mathrm{o} \qquad \mathbb{V}\left[y_*|\mathbf{x}_*\right] = k\left(\mathbf{x}_*, \mathbf{x}_*\right) - \overline{\mathbf{k}}_*^\mathrm{T}\overline{\mathbf{K}}_\mathrm{o}^{-1}\overline{\mathbf{k}}_* + \sigma_\mathrm{n}^2 \tag{33}$$

where the target vector now includes the observed derivatives:

$$\overline{\mathbf{t}}_\mathrm{o} = \begin{bmatrix}t_{\mathrm{o}1} & \cdots & t_{\mathrm{o}N} & \mathbf{t}'_{\mathrm{o}1} & \cdots & \mathbf{t}'_{\mathrm{o}N}\end{bmatrix}^\mathrm{T} \tag{34}$$

and predictions for the tangent are done as in Eq. (27) but now by differentiating $\overline{\mathbf{k}}_*$ instead of $\mathbf{k}_*$. Fig. 2 shows GP predictions for $t = \sin(x)$ with two training points with and without including derivative observations. Now the conditioning not only constrains the predictions to agree with the target values in $\mathcal{D}$ but also with their derivatives. This makes effective use of the limited amount of information coming from a small number of *online* observations, as making a single gradient observation is equivalent to adding D extra function observations around the target value.

3.5. Hyperparameter optimization

The process variance σ_f^2 and length scale ℓ that compose the kernel and the target noise σ_n^2 are hyperparameters that should be learned from the dataset $\mathcal{D}$. Since a full Bayesian treatment for these parameters — introducing a prior, deriving a posterior and marginalizing — usually demands the use of expensive numerical techniques such as Markov Chain Monte Carlo (MCMC), here we opt for a maximum likelihood solution in order to minimize the computational overhead associated with the *online* calibration of the GP models. Furthermore, due to the limited amount of data available for estimation, here we refrain from optimizing for the derivative noise σ_d^2 and instead assume derivative observations are noiseless.

The aim here is to maximize the marginal likelihood, obtained by averaging the probability that the model reproduces the training targets over all possible values of $\overline{\mathbf{y}}_\mathrm{o}$ associated with $\overline{\mathbf{t}}_\mathrm{o}$:

$$p\left(\overline{\mathbf{t}}_\mathrm{o}|\sigma_\mathrm{f}^2, \sigma_\mathrm{n}^2, \ell\right) = \int p\left(\overline{\mathbf{t}}_\mathrm{o}|\overline{\mathbf{y}}_\mathrm{o}, \sigma_\mathrm{n}^2\right) p\left(\overline{\mathbf{y}}_\mathrm{o}|\sigma_\mathrm{f}^2, \ell\right) \mathrm{d}\overline{\mathbf{y}}_\mathrm{o} \tag{35}$$

which is a function that depends only on the hyperparameters. We optimize this marginal likelihood with a BFGS algorithm [44], which requires the computation of the gradient of $p\left(\overline{\mathbf{t}}\right)$. Taking the natural logarithm of both sides of Eq. (35) and differentiating with respect to $\overline{\mathbf{K}}_\mathrm{o}$ yields an expression for the gradient:

$$\nabla \ln p\left(\overline{\mathbf{t}}_\mathrm{o}\right) = -\frac{1}{2}\mathrm{tr}\left(\overline{\mathbf{K}}_\mathrm{o}^{-1}\nabla\overline{\mathbf{K}}_\mathrm{o}\right) + \frac{1}{2}\overline{\mathbf{t}}_\mathrm{o}^\mathrm{T}\overline{\mathbf{K}}_\mathrm{o}^{-1}\nabla\overline{\mathbf{K}}_\mathrm{o}\overline{\mathbf{t}}_\mathrm{o} \tag{36}$$

and $\nabla\overline{\mathbf{K}}_\mathrm{o}$ can be obtained in a straightforward manner by differentiating Eqs. (19), (28) and (30) with respect to each hyperparameter and reassembling the matrix as in Eq. (31).

4. Surrogate modeling framework

In this section, we use the techniques of Section 3.2 to build an adaptive surrogate modeling framework for FE^2. Following Fig. 1, we introduce a surrogate model $\mathcal{S}$ trained on a set of constitutive observations $\mathcal{D}$ obtained from a small number of fully-solved *anchor* models (gathered in the set $\mathcal{A}$) subjected to the strain histories from their respective anchoring integration points. For bulk homogenization, we define $\mathcal{S}$ as:

$$\boldsymbol{\sigma}^{\Omega}\left(\boldsymbol{\varepsilon}^{\Omega},\mathcal{D}\right)=\mathbf{D}_{\mathrm{e}}^{\Omega}\boldsymbol{\varepsilon}^{\Omega}+\mathbb{E}\left[\widehat{\boldsymbol{\sigma}}^{\Omega}\left(\boldsymbol{\varepsilon}^{\Omega},\mathcal{D}\right)\right] \tag{37}$$

where $\mathbf{D}_{\mathrm{e}}^{\Omega}$ is an initial stiffness obtained from the first computed integration point and $\widehat{\boldsymbol{\sigma}}$ is a stress correction given by the mean GP response, with the variance being used exclusively for adaptivity purposes. The consistent tangent stiffness is the combination of $\mathbf{D}_{\mathrm{e}}^{\Omega}$ and the derivatives of the expected stress corrections (Eq. (27)). The dataset $\mathcal{D}$ is populated with stress and stiffness values coming from points with the highest values for the uncertainty over any component of $\widehat{\boldsymbol{\sigma}}$.

We opt for modeling each component of $\widehat{\boldsymbol{\sigma}}$ (three components for the 2D examples treated here, with which we implicitly enforce symmetry to the stress tensor) independently, each with its own zero-mean GP model. This strategy implicitly states that components of the stress tensor are a priori uncorrelated, which greatly simplifies the structure of the covariance matrix of the GP models and reduces the number of hyperparameters to be estimated. Furthermore, we assume for now that $\widehat{\boldsymbol{\sigma}}$ depends only on the current strain value $\boldsymbol{\varepsilon}^{\Omega}$, which means that the GP model loads, unloads and reloads along the same path. This is a limitation to be addressed in future versions of the framework that currently hinders its ability to treat general non-monotonic load paths.[1] It is worth mentioning, however, that path dependency is already partially accounted for since different strain histories will lead to the construction of different surrogates.

Note that the generality of the $\mathcal{S}$ model given by Eq. (37) is not compromised by the elastic-correction additive decomposition since $\widehat{\boldsymbol{\sigma}}$ can take any shape, but for initially linear-elastic materials this split improves the robustness of the surrogate in two ways. Firstly, it helps the GP in reproducing the initial elastic behavior at a moment when the surrogate models have very little data to work with. Secondly, it aids in preventing the occurrence of spurious strain localization as the GP moves back to its zero prior: away from the training points, the model of Eq. (37) returns instead to a linear-elastic response.

At this point, it is relevant to distinguish the present approach from other interesting active learning strategies for constitutive modeling proposed in literature [45]. In [31,32], the authors construct a constitutive database from which observations are adaptively selected and used to build surrogate GP constitutive models both for stresses and for a number of assumed macroscopic internal variables, creating a one-to-one mapping between inputs and outputs. Here we opt for a more general approach and do not assume the existence of macroscopic internal variables. This choice is motivated by the basic FE^2 premise that microscopic behavior is too complex to conform to an *a priori* assumed macroscopic constitutive model and leads to a non-unique mapping involving exclusively stresses and strains. This effectively amounts to a reduction in dimensionality of the feature space of the surrogates that allows us to employ the same framework for completely distinct constitutive models (*c.f.* Eqs. (37) and (43)). This dimensionality reduction implicitly creates the need for the existence of anchor models and for subjecting them to the strain paths seen by their respective anchoring points, allowing the microscopic internal variables — which are not observed by the surrogate — to influence the macroscopic stress-strain response and compensate for the loss of information incurred when avoiding making assumptions on macroscopic internal variables.[2]

In order to position the present active learning approach within a general FEM implementation, it is useful to recall the main steps involved in finding equilibrium solutions for nonlinear finite element models. These are shown in Fig. 3. A solution for $\mathbf{u}$ is obtained iteratively by minimizing a global force residual $\mathbf{r}$ computed from the material response at every integration point (`materialUpdate`). Upon convergence and before moving to the next time step, the current solution is checked (`checkSolution`) and can be rejected if there is a need to adapt the model (*e.g.* nucleate/propagate cracks, refine the mesh, change constitutive models). Once the solution is converged and accepted, material history is updated (`commit`) and the model moves to the next time step.

In an FE^2 model, Fig. 3 can represent the macroscopic solution loop and the micromodels can be seen as a material-like entity, which leads to another similar solution loop being embedded within the `materialUpdate` routine. Here we exploit this modularity and discuss the implementation of the learning approach as a surrogate to an arbitrary material model denoted as `fullModel`. We can therefore implement the approach as a material wrapper that encapsulates `fullModel` and handles learning and prediction tasks. Algorithms 1, 2, 4 and 5 show how the material routines marked in bold in Fig. 3 are implemented for this wrapper. In the following, we elaborate on each of these components and present the main implementational aspects of the framework in detail.

[1] This can be addressed by augmenting the input space, the most straightforward way being using both $\boldsymbol{\varepsilon}^{\Omega}$ and $\Delta\boldsymbol{\varepsilon}^{\Omega}$ as inputs to the GP models [33,34].

[2] The noise hyperparameter σ_{f}^2 learned during the analysis implicitly accounts for the dimensionality reduction error incurred by assuming two anchor models operating at the same strain level but with different microscopic internal states have the same stress response.

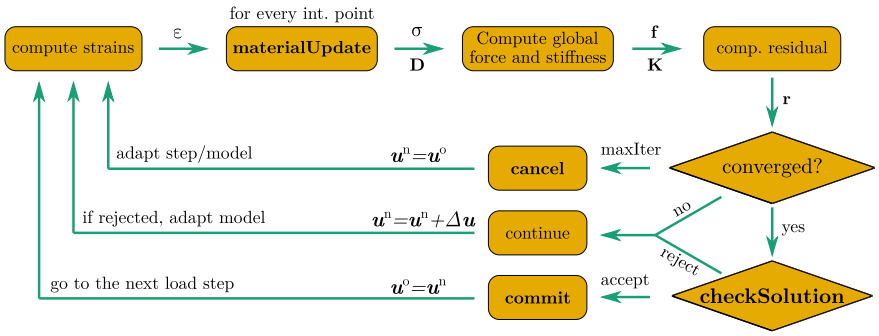


Fig. 3. Schematic analysis flow for finite element problems involving nonlinear material behavior. The subscripts o and n refer to old (converged) and new (current) values, respectively. The proposed acceleration framework focuses on the steps marked in bold.

4.1. Initial sampling

Since there is no *offline* training, the surrogate model starts with no prior information on the constitutive behavior being approximated ($\mathcal{D} = \varnothing$) and no *anchor* models initially present ($\mathcal{A} = \varnothing$). We therefore introduce an initialization step by solving the first time increment under the assumption that all integration points are deforming elastically in order to obtain an initial strain distribution which we use to initialize $\mathcal{A}$ and $\mathcal{D}$. During this step, we use the fully-solved model only once in order to obtain the initial stiffness $\mathbf{D}_e$ and we assume all other points have the same stiffness in order to avoid further full-order computations (Algorithm 1). As we will see shortly, this is not a limiting assumption since this first elastic approximation is rejected and the first time step is revisited after the GP models and initial anchors are initialized.

Algorithm 1: The `materialUpdate` routine.

Input: strain $\boldsymbol{\varepsilon}$ at the integration point p
Output: stress $\boldsymbol{\sigma}$ and stiffness $\mathbf{D}$ at the integration point

if *initialization step* **:**
  if $\mathbf{D}_e$ *is not initialized (i.e. p is the first point ever computed)* **:**
    use original constitutive model: $(\boldsymbol{\sigma}, \mathbf{D}) \leftarrow$ `fullModel::materialUpdate`$(\boldsymbol{\varepsilon})$;
    initialize elastic stiffness: $\mathbf{D}_e \leftarrow \mathbf{D}$;
  else
    assume p has the same stiffness as the first computed point: $\boldsymbol{\sigma} \leftarrow \mathbf{D}_e\boldsymbol{\varepsilon}$; $\mathbf{D} \leftarrow \mathbf{D}_e$;
else
  compute elastic stresses: $\boldsymbol{\sigma}_e \leftarrow \mathbf{D}_e\boldsymbol{\varepsilon}$;
  use GP to predict a constitutive correction: $\left(\mathbb{E}\left[\widehat{\boldsymbol{\sigma}}, \widehat{\mathbf{D}}\right], \mathbb{V}\left[\widehat{\boldsymbol{\sigma}}\right]\right) \leftarrow \mathrm{GP}(\boldsymbol{\varepsilon})$;
  approximate the response: $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}_e + \mathbb{E}\left[\widehat{\boldsymbol{\sigma}}\right]$, $\mathbf{D} \leftarrow \mathbf{D}_e + \mathbb{E}\left[\widehat{\mathbf{D}}\right]$;
  compute the uncertainty indicator γ as in Eq. (38);
  if $\gamma > \gamma_{\text{cancel}}$ **:**
    cancel time step (Algorithm 5);
  if *softening is detected ($D_{ii} < 0$)* **:**
    penalize point uncertainty: $\gamma \leftarrow \gamma - D_{ii}$;
  if *last step has been cancelled* **:**
    switch to a modified Newton-Raphson strategy: $\mathbf{D} \leftarrow \mathbf{D}_e$;
  store values for this point: $\boldsymbol{\varepsilon}_p^n \leftarrow \boldsymbol{\varepsilon}$; $\gamma_p^n \leftarrow \gamma$;
return $\boldsymbol{\sigma}, \mathbf{D}$

After this first approximation for the solution is obtained, the `checkSolution` routine of Algorithm 2 is called. Since the strains at every integration point are stored, an informed initial choice for $\mathcal{A}$ can be made by clustering the integration points using a k-means clustering algorithm [46]. For each cluster, the point closest to the cluster centroid is chosen, added to $\mathcal{A}$ and immediately sampled. The GP models (one for each stress component) are then prepared to make predictions by computing and storing factorized versions of their covariance matrices.

Initial values for the hyperparameters σ_f^2, σ_n^2 and ℓ can either be provided by the user based on prior knowledge (*e.g.* from using the same material on other models)[3] or be estimated *online* with an initial training procedure. We do this here by creating a fictitious anchor model for each cluster and loading it monotonically in the strain direction seen by the

[3] Hyperparameters reflect intrinsic patterns in the constitutive manifold being inferred and should therefore be fairly insensitive to the macroscopic model being solved.

Algorithm 2: The `checkSolution` routine.

if *first time step* **:**
  initialize dataset and anchor point set: $\mathcal{D} \leftarrow \varnothing$, $\mathcal{A} \leftarrow \varnothing$;
  divide the integration points into k clusters in strain space: $\left(\boldsymbol{\mathcal{C}}, \overline{\boldsymbol{\varepsilon}}^{\mathrm{n}}\right) \leftarrow \mathtt{KMC}\left(\boldsymbol{\varepsilon}^{\mathrm{n}}, k\right)$;
  for *every cluster* $\mathcal{C}_i \in \boldsymbol{\mathcal{C}}$ **:**
    find representative point $p = \underset{j \in \mathcal{C}_i}{\arg\min} \left\| \boldsymbol{\varepsilon}_j^{\mathrm{n}} - \overline{\boldsymbol{\varepsilon}}_i^{\mathrm{n}} \right\|$;
    anchor a fully-solved model at point p: $\mathcal{A} \leftarrow \mathcal{A} \cup p$;
    solve the newly-created full model once: $(\boldsymbol{\sigma}, \mathbf{D}) \leftarrow \mathtt{fullModel} :: \mathtt{materialUpdate}\left(\boldsymbol{\varepsilon}_p^{\mathrm{n}}\right)$;
    compute corrections and add data: $\widehat{\boldsymbol{\sigma}} \leftarrow \boldsymbol{\sigma} - \mathbf{D}_{\mathrm{e}} \boldsymbol{\varepsilon}_p^{\mathrm{n}}$; $\widehat{\mathbf{D}} \leftarrow \mathbf{D} - \mathbf{D}_{\mathrm{e}}$; $\mathcal{D} \leftarrow \mathcal{D} \cup \left(\boldsymbol{\varepsilon}_p^{\mathrm{n}}, [\widehat{\boldsymbol{\sigma}} \quad \widehat{\mathbf{D}}]\right)$;
  update GP and store marginal likelihood: $\mathtt{updateGP}(\mathcal{D})$; $L \leftarrow \ln p\,(\mathbf{y}|\mathbf{x})$;
  reject solution (revisit the first time step);
else
  update $\mathcal{D}$ based on the tolerance γ_{tol} using Algorithm 3: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathtt{addData}\,(\gamma_{\mathrm{tol}})$;
  if $\mathcal{T}$ *has not changed* **:**
    accept solution as is (Algorithm 4);
  else
    refactor covariance matrix: $\mathtt{updateGP}(\mathcal{D})$;
    if $\left\| L / \ln p\,(\overline{\mathbf{t}}) \right\| > L_{\mathrm{retrain}}$ **:**
      recompute GP hyperparameters: $\mathtt{retrainGP}(\mathcal{D})$; $L \leftarrow \ln p\,(\overline{\mathbf{t}})$;
    reject solution (continue on the same time step);

point closest to the cluster centroid. In order to avoid redundancy (*e.g.* adding linear-elastic data coming from multiple models), we first initialize the GP with the standard hyperparameter values $\sigma_{\mathrm{f}}^2 = 1.0$ MPa^2, $\ell = 1.0 \times 10^{-2}$, $\sigma_{\mathrm{n}}^2 = 0.0$ MPa^2 and use a variance threshold as a dosing mechanism to ensure the added data is uniformly spaced. After optimizing for the hyperparameters, we discard the data obtained from the fictitious anchors since there is no guarantee any of the real integration points will follow the same strain history. Once $\mathcal{D}$ and $\mathcal{A}$ are initialized, the first time step is then revisited by rejecting the initial elastic approximation done in Algorithm 1. Note that this happens only once at the beginning of the analysis.

4.2. Active learning

After the initialization step, the GP surrogates are used to compute $\widehat{\boldsymbol{\sigma}}$ and the response is approximated as in Eq. (37) (refer to Fig. 3 for the analysis flow of a single load increment). Since an independent GP model is used for each stress component, we adopt a single uncertainty indicator γ given by:

$$\gamma = \max_{i}^{n_\sigma} \left(\sqrt{\mathbb{V}\left[\widehat{\sigma}_i\right]} \right) \tag{38}$$

with n_σ being the number of stress components, and use it to drive the adaptive components of the framework. The indicator γ is therefore an absolute measure of uncertainty (expressed in units of stress). We empirically find this choice of indicator to offer the best performance in the examples treated in this work. Nevertheless, other alternatives might perform better depending on the problem being solved – *e.g.* normalizing $\mathbb{V}\left[\widehat{\sigma}_i\right]$ with the process variance, which for the present examples we find to adversely affect the performance of the framework when σ_{f}^2 is periodically re-estimated. The literature on active learning and greedy GP approximations can also provide the interested reader with further alternatives for defining γ (refer for instance to Section 8.3.3 of [41] and references within).

Values of γ for all integration points are updated at every global Newton-Raphson iteration (Algorithm 1) and their final values $\boldsymbol{\gamma}^{\mathrm{n}}$ (upon global convergence) are used to drive a greedy refinement of the GP models (Algorithm 2) and ensure the surrogate model remains accurate. This is enforced by the tolerance parameter γ_{tol} through the `addData` routine of Algorithm 3. Here we define an additional set $\mathcal{T}$ of tracked anchors that contains the models for which data has been added during the present time step. The set $\mathcal{T}$ is used both to avoid repeatedly adding data from a single anchor model at any given time step and to make sure the single added data point is updated to reflect the latest equilibrium solution (since from Fig. 3, multiple `materialUpdate` → `checkSolution` → `continue` cycles can occur before history is committed).

The uncertainty level γ is checked at every integration point and the anchor model associated with the point having the highest value is chosen for sampling. In order to keep the number of models in $\mathcal{A}$ to a minimum, `addData` gives priority to models already in $\mathcal{A}$ and only considers other potential anchoring points if every model in $\mathcal{A}$ with an uncertainty higher than γ_{tol} has already been sampled on the current time step. Since the macroscopic problem being solved imposes a degree of similarity between integration points, the idea is that adding data from points in $\mathcal{A}$ should also help reduce the uncertainty of nearby integration points. This can therefore be seen as a *greedy* data selection approach [41] that aims at keeping both $\mathcal{A}$ and $\mathcal{D}$ as small as possible. If all models in $\mathcal{A}$ have already been sampled and a new anchoring point must be chosen, we first recover the material history of the newly-created model by making it revisit the complete cumulative strain history $\boldsymbol{\varepsilon}_{\mathrm{h}}^{\overline{p}}$ of its anchoring point $\overline{p}$. Also note that we avoid adding data from models in $\mathcal{A}$ undergoing unloading or

reloading. This is a consequence of assuming a unique relationship between strains and stresses: data coming from anchor models unloading through a different path would be erroneously interpreted by the GP models as noise.

Algorithm 3: The `addData` routine.

Input: A tolerance γ_{tol}
Output: A single data point to be included in $\mathcal{D}$

update data sampled from $\mathcal{T}$ to reflect the new equilibrium strains $\boldsymbol{\varepsilon}^{\text{n}}_{p\in\mathcal{T}}$;
update GP models and uncertainty indicators: `updateGP`$(\mathcal{D})$; $\boldsymbol{\gamma}^{\text{n}} \leftarrow \boldsymbol{\gamma}^{\boldsymbol{\varepsilon}^{\text{n}}}$;
try to find an untracked anchor point to sample: $\overline{p} = \underset{p\in\mathcal{A}, p\notin\mathcal{T}}{\arg\max}\ \gamma^{\text{n}}_p | \gamma^{\text{n}}_p > \gamma_{\text{tol}}$;
if $\overline{p}$ *cannot be found* **:**
  try to find a new anchoring point: $\overline{p} = \underset{p\notin\mathcal{A}, p\notin\mathcal{T}}{\arg\max}\ \gamma^{\text{n}}_p | \gamma^{\text{n}}_p > \gamma_{\text{tol}}$;
update list of tracked points: $\mathcal{T} \leftarrow \mathcal{T} \cup \overline{p}$;
update list of anchor points: $\mathcal{A} \leftarrow \mathcal{A} \cup \overline{p}$ if $\overline{p} \notin \mathcal{A}$;
call `fullModel` :: `materialUpdate`$(\boldsymbol{\varepsilon}^{\text{h}}_{\overline{p}})$ to gradually bring the anchor at $\overline{p}$ to the current time step;
compute model response at $\boldsymbol{\varepsilon}^{\text{n}}_{\overline{p}}$: $(\boldsymbol{\sigma}, \mathbf{D}) \leftarrow$ `fullModel` :: `materialUpdate`$\left(\boldsymbol{\varepsilon}^{\text{n}}_{\overline{p}}\right)$;
compute corrections: $\widehat{\boldsymbol{\sigma}}^{\text{n}}_{\overline{p}} \leftarrow \boldsymbol{\sigma} - \mathbf{D}_{\text{e}}\boldsymbol{\varepsilon}^{\text{n}}_{\overline{p}}$; $\widehat{\mathbf{D}}^{\text{n}}_{\overline{p}} \leftarrow \mathbf{D} - \mathbf{D}_{\text{e}}$;
return $\left(\boldsymbol{\varepsilon}^{t}_{\overline{p}}, \left[\widehat{\boldsymbol{\sigma}}^{t}_{\overline{p}} \quad \widehat{\mathbf{D}}^{t}_{\overline{p}}\right]\right)$ with t being the latest step on which $\overline{p}$ is not unloading/reloading;

Upon making changes to $\mathcal{D}$, the covariance matrices of all GP models are updated and refactored. The addition of a new observation also leads to a change in marginal likelihood (Eq. (35)). We therefore allow for a re-estimation of the hyperparameters to take place once the likelihood reaches a value L_{retrain} times lower than the one computed after the latest estimation. If $\mathcal{T}$ has not changed during the latest call to `addData`, the current solution is accepted as it is. Otherwise we continue with the same time step by rejecting the current solution, which will cause the global Newton-Raphson solver to keep searching for an equilibrium solution (refer to Fig. 3) but now with updated GP models.

Two additional adaptivity safeguards are put in place in the `materialUpdate` routine of Algorithm 1. Firstly, the time step can be canceled if the uncertainty is higher than a threshold γ_{cancel} in order to avoid considering equilibrium solutions that are excessively far from the constitutive behavior being approximated. As we will discuss in Section 4.3, canceling the current load step does not mean giving up on the analysis, with new solution attempts being made after including extra anchor models in $\mathcal{A}$ and retraining the GP models. Secondly, the diagonal of the tangent stiffness matrix is checked for possible negative values, providing an indication of softening. This would indicate a switch from stresses to tractions is in order, but the decision for such a constitutive model switch should always be based on accurate information obtained from models in $\mathcal{A}$. Therefore, we flag the point for sampling by penalizing its uncertainty.[4]

Apart from the very first update computed in order to obtain $\mathbf{D}_{\text{e}}$, the expensive full-order model is never computed during `materialUpdate` (Algorithm 1). An alternative to this approach would be to actually call the models in $\mathcal{A}$ every time stresses at the anchoring points are computed. We do not opt for this alternative for two reasons. Firstly, using a single constitutive model (the surrogate $\mathcal{S}$) for the whole mesh avoids potential non-uniqueness issues that would arise, for instance, if points in $\mathcal{A}$ are switching between different constitutive regimes (loading/unloading/softening). Secondly, refraining from constantly updating the models in $\mathcal{A}$ results in significant gains in terms of acceleration by making the reduced model insensitive to the number of global Newton-Raphson iterations needed for convergence and allowing full-order models in $\mathcal{A}$ to become dormant and essentially be removed from the analysis for as long as the uncertainty at the associated anchoring point does not increase. However, it is worth mentioning in passing that the more expensive alternative has the merit of allowing for an extra *novelty detection* safeguard to be employed through which the deviation between predictions for $\boldsymbol{\sigma}$ coming from the full-order and surrogate models can be kept in check.

4.3. Solution robustness

Algorithms 2 and 3 ensure that only a single data point is added to $\mathcal{D}$ at a time. This is done in order to avoid large perturbations to the current equilibrium solution caused by changes in the constitutive response.[5] Once a new observation is added, the current (now unconverged) solution is kept but the algorithm continues on the same time step until equilibrium is reestablished, at which point `checkSolution` is once again called (refer to Fig. 3 for the order in which these routines are called within a single analysis step). This process is repeated until γ^{n} is lower than γ_{tol} everywhere. The resulting cycle of carefully adding data without significantly drifting away from equilibrium helps keeping the global solution scheme robust.

[4] In the examples of this paper we do not treat models for which such a switch from bulk to cohesive behavior would be necessary. The safeguard is therefore only triggered in rare occasions when the GP is trying to approximate a perfectly-plastic response.

[5] Adding data to the GP affects the response of all points within a hypersphere in strain space with a radius that depends on ℓ. The resulting change in global response may cause the solver to diverge.

Algorithm 4: The `commit` routine.

clear list of tracked models: $\mathcal{T} \leftarrow \varnothing$;
store converged values: $\boldsymbol{\gamma}^{\mathrm{o}} \leftarrow \boldsymbol{\gamma}^{\mathrm{n}}$;
update persistent strain history: $\boldsymbol{\varepsilon}^{\mathrm{h}} \leftarrow \left[\, \boldsymbol{\varepsilon}^{\mathrm{h}} \quad \boldsymbol{\varepsilon}^{\mathrm{n}} \,\right]$;
go to next time step;

When the solution converges and $\gamma^{\mathrm{n}} < \gamma_{\mathrm{tol}}$ for all integration points, the `commit` routine is called (Algorithm 4). Before moving to the next step, converged values for $\boldsymbol{\gamma}$ are updated. If the global solver fails to converge or if the additional uncertainty threshold γ_{cancel} of Algorithm 1 is violated, the solution for the current time step is canceled (Algorithm 5). Before making a new solution attempt, converged values are recovered and Algorithm 3 is used to sample the model with highest uncertainty even if its value for γ is lower than γ_{tol}. Furthermore, we switch to a secant solution strategy by fixing the stiffness matrix to $\mathbf{D}_{\mathrm{e}}$ (Algorithm 1). We empirically notice that, for certain combinations of hyperparameters, the surrogate model causes the global Newton-Raphson solver to lose robustness due to rapid changes in stiffness. Switching to a secant strategy after a canceled step assures a solution is found under this scenario, albeit with a lower convergence rate.

Algorithm 5: The `cancel` routine.

recover converged values: $\boldsymbol{\gamma}^{\mathrm{n}} \leftarrow \boldsymbol{\gamma}^{\mathrm{o}}$;
add data from the point with the highest variance (Algorithm 3): $\mathcal{D} \leftarrow \mathcal{D} \cup \texttt{addData}\,(0)$;
refactor covariance matrix: $\texttt{updateGP}\,(\mathcal{D})$;
if $\left\| L / \ln p\left(\bar{\mathbf{t}}\right) \right\| > L_{\mathrm{retrain}}$ **:**
  recompute GP hyperparameters: $\texttt{retrainGP}\,(\mathcal{D})$; $L \leftarrow \ln p\left(\bar{\mathbf{t}}\right)$;
go back to the beginning of the time step;

5. Numerical examples

In this section, we put the proposed framework to the test on a number of numerical examples. The algorithms of Section 4 have been implemented in an in-house Finite Element code using the open-source Jem/Jive C++ numerical analysis library [47]. We begin by demonstrating the applicability of the framework for FE^2 analysis and move on to performing an in-depth investigation of its performance with fast single-scale homogeneous examples that allow for extensive parametric studies to be performed. It is emphasized that, although it is our vision to use the surrogate model in a multiscale context, it does not matter for the purpose of testing the active learning framework whether the material update of the full model involves solving a micromechanical BVP or just evaluating a nonlinear constitutive relation.

5.1. FE^2 demonstration

The first example concerns a fiber-reinforced composite tapered specimen loaded in transverse tension. The geometry, mesh and boundary conditions are shown in Fig. 4. A 4-fiber RVE model is embedded at each macroscopic integration point. The geometry and mesh of the micromodel are also shown in Fig. 4. The fibers are modeled as linear elastic with properties $E = 74000$ MPa and $\nu = 0.2$. For the matrix we employ the pressure-dependent elastoplastic model proposed by Melro *et al.* [48], with $E = 3130$ MPa, $\nu = 0.37$, $\nu_{\mathrm{p}} = 0.32$ (plastic Poisson's ratio) and yield stresses given by:

$$\sigma_{\mathrm{t}} = 64.80 - 33.6\mathrm{e}^{-\varepsilon^{\mathrm{p}}_{\mathrm{eq}}/0.003407} - 10.21\mathrm{e}^{-\varepsilon^{\mathrm{p}}_{\mathrm{eq}}/0.06493} \tag{39}$$

$$\sigma_{\mathrm{c}} = 81.00 - 42.0\mathrm{e}^{-\varepsilon^{\mathrm{p}}_{\mathrm{eq}}/0.003407} - 12.77\mathrm{e}^{-\varepsilon^{\mathrm{p}}_{\mathrm{eq}}/0.06493} \tag{40}$$

where $\varepsilon^{\mathrm{p}}_{\mathrm{eq}}$ is the equivalent plastic strain. The model is solved for 100 load steps, at which point the global macroscopic response is almost perfectly plastic and the strain localizes around the center of the specimen.

We run the reduced model with $k = 1$, $\gamma_{\mathrm{tol}} = 0.3$ MPa, $\gamma_{\mathrm{cancel}} = 20$ MPa and with hyperparameters estimated using a single micromodel loaded in the direction of the single k-means clustering centroid (in this case the average strain in the specimen). The analysis starts with $\mathcal{A} = \varnothing$ and $\mathcal{D} = \varnothing$ and ends with a total of 14 anchor models in $\mathcal{A}$ (out of a total of 134 points) and $|\mathcal{D}| = 73$ data points.

We plot the resultant load-displacement response for both full- and reduced-order models in Fig. 5. The active learning approach is able to capture the correct global model response with negligible loss of accuracy while running 22 times faster than the full-order model. Of the 35 s execution time of the reduced model, a total of 15 s is spent on updating the GP models and using them for new predictions. For larger micromodels with denser meshes, the execution time of the reduced model will be dominated by computing the few micromodels included in $\mathcal{A}$ and the overhead associated with the GP models will become negligible. For the full-order model, 99% of the execution time is spent solving the embedded micromodels, confirming the bottleneck assumption of Eq. (8). In Fig. 6 we plot the horizontal stress distribution along

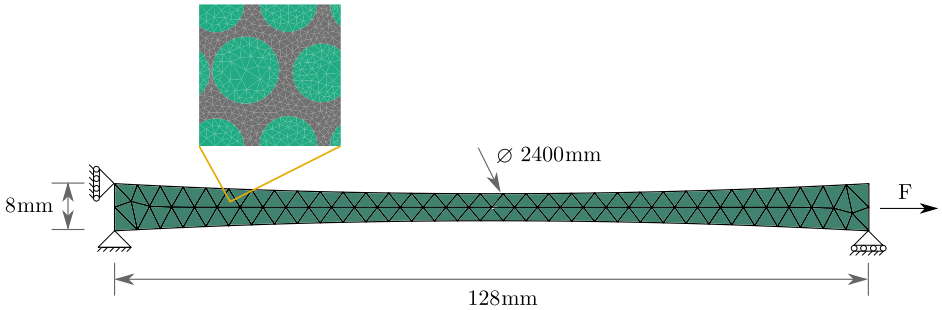


Fig. 4. FE^2 demonstration: A composite tapered bar loaded in transverse tension.

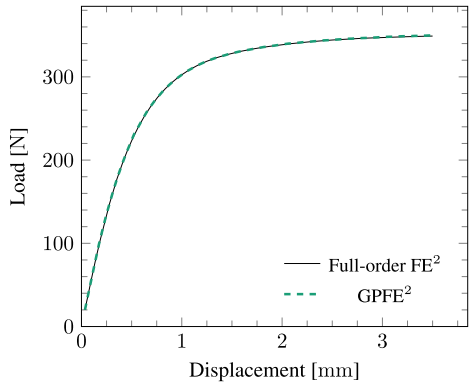


Fig. 5. Load-displacement curves obtained with the full-order FE^2 approach and with our reduction framework. The reduced model runs 22 times faster with only negligible loss of accuracy.

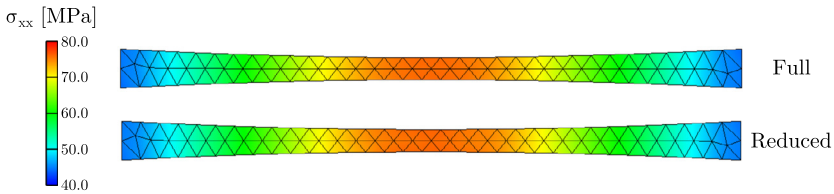


Fig. 6. FE^2 tapered bar: Stress fields at the end of the analysis for both full-order and reduced models. Our active learning approach predicts a stress distribution indistinguishable from the full-order one.

the specimen at the last time step for both models. No discernible differences between the two stress distributions can be seen.

5.2. Performance and parametric sensitivity

The implementation of Section 4 allows for the active learning framework to supplant any full-order constitutive model. In the example of Section 5.1, this full-order model is the embedded RVE model of Fig. 4. We now switch to a single-scale model with the homogeneous and elastoplastic material used in the previous example. This allows for an in-depth investigation on the performance of the reduction framework to be performed without loss of generality and without resorting to running a large number of expensive FE^2 simulations.

We start the investigation by further simplifying the macroscopic model to the one-dimensional bar with variable cross-section area shown in Fig. 7, reducing the dimensionality of the constitutive space being approximated to only two dimensions ($\varepsilon_{xx} - \sigma_{xx}$). This change allows for the full model being inferred to be easily visualized and reduces the number of hyperparameters from nine for the two-dimensional case to only three.[6] The original tapered geometry is simulated by making the cross-sectional area of the bar depend on the x coordinate:

[6] For each stress component we have three hyperparameters to be determined, namely σ_f^2, σ_n^2 and ℓ.

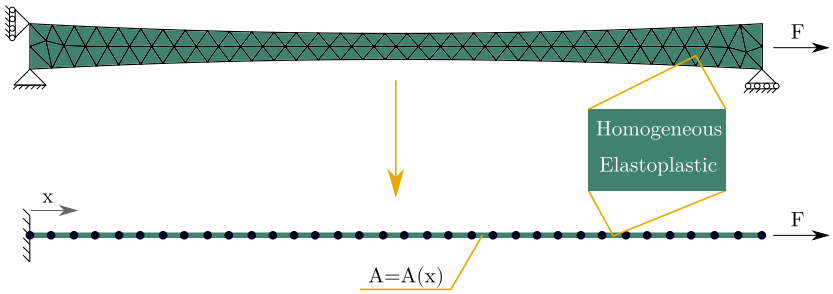


Fig. 7. Homogeneous models used to study the parametric sensitivity of the framework. The additional simplification to a bar problem allows for the constitutive space to be visualized in two dimensions (ε_{xx}-σ_{xx}).

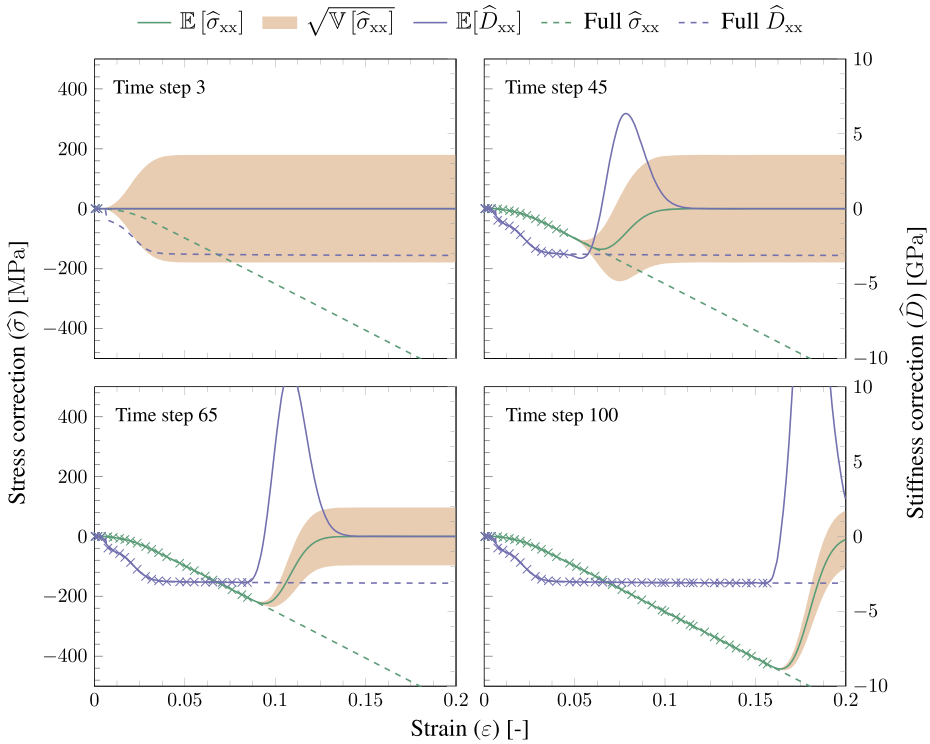


Fig. 8. Evolution of the GP constitutive model as data is gradually added. Dashed lines show the full-order response being approximated. The observations in $\mathcal{D}$ are plotted as crosses.

$$A(x) = 0.8 - 2.0\left(0.0534x - 0.000418x^2\right) \tag{41}$$

5.2.1. Evolution of the GP regression

We run the reduced model with $k = 1$, $\gamma_{\text{tol}} = 0.4$ MPa, $\gamma_{\text{cancel}} = 20$ MPa and $L_{\text{retrain}} = 10$, with hyperparameter values obtained by loading a single material point in the k-means centroid direction. We can visualize the gradual improvement of the surrogate model as data is added by plotting the GP predictions together with the exact constitutive response being approximated at different moments throughout the analysis. This can be seen in Fig. 8, which shows snapshots made at four different time steps.

As we start with an empty dataset, at first the GP model predicts linear-elastic behavior for the complete strain range, but with an uncertainty that quickly increases away from the observations (marked as crosses in Fig. 8). This increase is the mechanism that triggers the sampling of extra information from models in $\mathcal{A}$. As more data is added to $\mathcal{D}$, the GP model is gradually refined and is able to reproduce the target full-order response with excellent accuracy, although we observe that even though the perfectly plastic response has a simple and constant shape, the GP is never able to extrapolate it. Away from the sampling points the GP predictions will always return to the prior with zero mean. By the end of the analysis, four models are present in $\mathcal{A}$ and $|\mathcal{D}| = 39$.

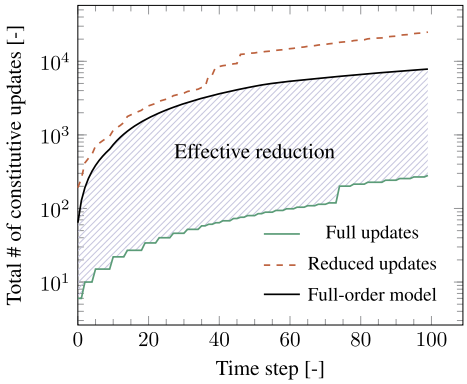


Fig. 9. Cumulative number of constitutive updates obtained with the full-order and reduced models. The shaded region represents the effective acceleration associated with the reduction framework.

5.2.2. Reduction ratio

In Section 2 we argued that the computational bottleneck of FE^2 lies on macroscopic constitutive model evaluations (Eq. (8)). We can therefore have an indication of the acceleration promoted by the active learning approach simply by counting the number of full-order model evaluations. We plot in Fig. 9 the evolution of the cumulative number of constitutive updates for the example of the previous section together with the number of updates obtained by using the full-order model at every integration point.

We see that the total number of material updates performed by the reduced model is significantly higher than that of the reference full-order model. Two different reasons for this increase can be identified. Firstly, four time step cancels occur after which the model switches to a secant approach that requires more iterations for convergence. Secondly, extra iterations are triggered every time a new observation is added to $\mathcal{D}$ and the solution deviates from equilibrium (Algorithms 2 and 3).

However, given that the bottleneck assumption of Eq. (8) holds, the effective acceleration brought by the adaptive reduction approach is only related to the number of times the anchor models are computed, and is therefore related to the shaded area shown in Fig. 9. Here we define R as the ratio between full model evaluations of the reference (full-order) and reduced-order models:

$$R = \frac{n^{\mathrm{f}}_{\mathrm{ref}}}{n^{\mathrm{f}}_{\mathrm{GP}}} \tag{42}$$

and use it as a measure of acceleration. For the model of Fig. 9, the ratio at the end of 100 time steps is $R = 27.9$.

5.2.3. Influence of γ_{tol}

The uncertainty tolerance γ_{tol} is the main parameter controlling the active learning procedure: lower values of γ_{tol} should lead to a higher sampling frequency while higher values should lead to smaller cardinalities for $\mathcal{D}$ and $\mathcal{A}$ at the cost of solution accuracy. In this section we put these claims to the test and investigate how much control can actually be exerted over the solution algorithm by changing γ_{tol}.

Going back to the one-dimensional example of the previous section, we solve the problem for multiple values of γ_{tol} between 0.3 MPa and 10.0 MPa. In Fig. 10 we plot the evolution of the cardinality of the dataset $\mathcal{D}$ for six different γ_{tol} values. We see that γ_{tol} indeed influences the number of observations added to $\mathcal{D}$, but only to a limited extent. This is due to the presence of the remaining adaptive components of the framework, namely the time step cancelling mechanism related to γ_{cancel}, the sampling of data with zero threshold after a canceled step and the hyperparameter retraining procedure linked to L_{retrain}. This combination of safeguards leads to datasets of similar sizes for most of the γ_{tol} values adopted here. It is interesting to note how the curve for $\gamma_{\mathrm{tol}} = 0.3$ MPa sharply increases in slope after time step 40, a moment when the hyperparameters are reoptimized and generate a modified model in which γ_{tol} is crossed more often. This indicates that the γ_{tol} level necessary to achieve a given sampling frequency is also influenced by the hyperparameter values.

Similar observations can be made by looking at the degree of control that can be exerted over the acceleration level provided by the framework, which we quantify through the reduction ratio R of Eq. (42). For each model we compute the evolution of the reduction ratio R with the time steps and plot them in Fig. 11a. During the first time steps, γ_{tol} has the expected influence on the acceleration level, with ratios as high as 150 being obtained for $\gamma_{\mathrm{tol}} = 10.0$ MPa. When new data is sampled, which requires a number of full-order computations to be performed, the reduction ratio experiences drops that become sharper for higher values of γ_{tol}. From time step 40, as the model approaches a perfectly-plastic regime and the sampling frequency increases (see Fig. 10), all models converge to reduction ratios between 12 and 30. Adjusting γ_{tol} in order to achieve a desired acceleration level is therefore not possible. Indeed, using higher values lead to lower accuracy — as can be seen in the load-displacement curves of Fig. 11b — without consistent gains in efficiency.

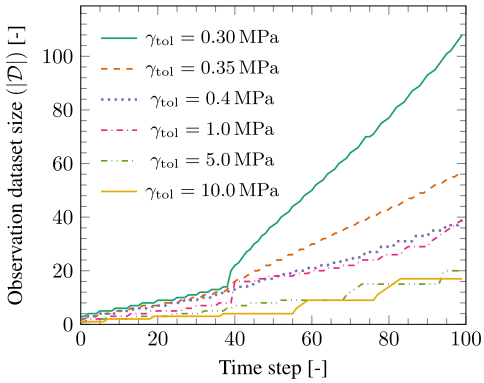


Fig. 10. Evolution of the cardinality of $\mathcal{D}$ for different values of γ_{tol}. A stricter uncertainty tolerance leads to a high sampling frequency.

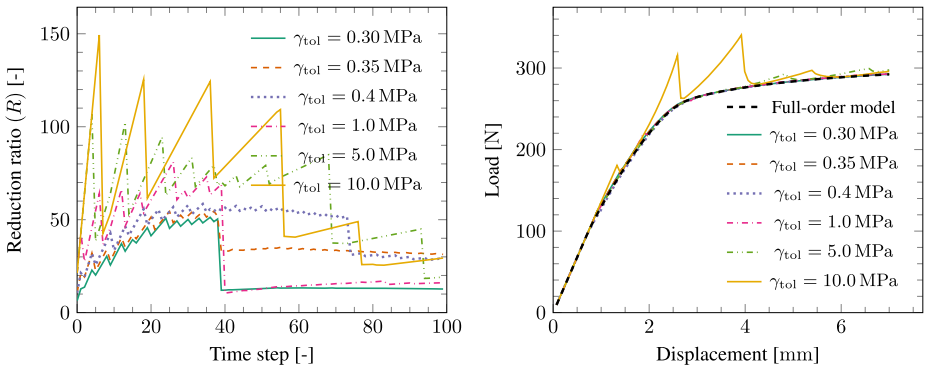


Fig. 11. Evolution of the reduction ratio and the number of fully-solved steps for models with different values of γ_{tol}. Model efficiency cannot be directly controlled by the uncertainty tolerance.

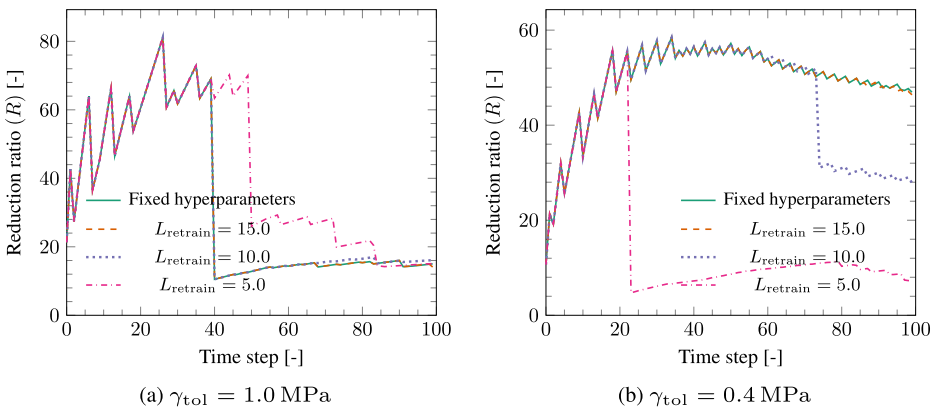


Fig. 12. Acceleration performance when using the hybrid constitutive model to solve the one-dimensional bar problem for different values of L_{retrain} and γ_{tol}.

5.2.4. Effect of re-estimating the hyperparameters

Up until this point, the hyperparameters σ_f^2, σ_n^2 and ℓ have been estimated at the beginning of the analysis and updated with a log marginal likelihood ratio threshold $L_{\text{retrain}} = 10$. For the next example, we return to the one-dimensional model of the previous sections but now using different values of L_{retrain}. Since the sampling frequency dictated by γ_{tol} is also influenced by the hyperparameters, we show results for two different values of γ_{tol}. The evolution of the reduction ratio R for these eight models is shown in Fig. 12.

Here we see two distinct behaviors depending on the uncertainty threshold level. For $\gamma_{\text{tol}} = 1.0$ MPa, allowing for a higher hyperparameter retraining frequency leads to higher acceleration factors up to time step 80. For the lower value of $\gamma_{\text{tol}} = 0.4$ MPa, retraining the hyperparameters has a detrimental effect on efficiency by leading to a much higher sampling frequency, with this change in behavior happening earlier for lower values of L_{retrain}. These results suggest there is no clear

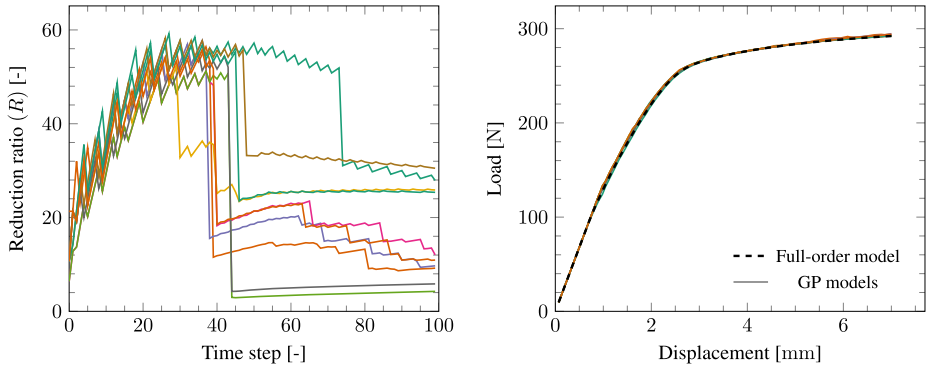


Fig. 13. Reduction ratios and load-displacement curves for 10 models with different initial hyperparameter values ($\gamma_{\text{tol}} = 0.4$ MPa, $L_{\text{retrain}} = 10.0$). A large spread in performance can be observed.

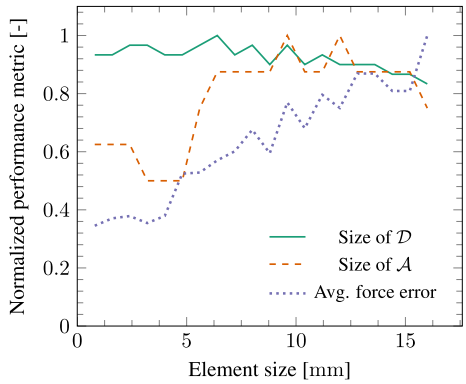


Fig. 14. Performance of the reduction framework for different levels of mesh discretization (two-dimensional bar problem). The amount of constitutive information necessary to build an accurate surrogate is independent of the total number of integration points in the model.

recommendation to be made on the optimum hyperparameter retraining frequency, as model performance is dictated by a complex interaction between γ_{tol}, the multiple adaptive model components and the hyperparameter values.

We show one last example before returning to the two-dimensional version of the model. We now keep both the retraining threshold $L_{\text{retrain}} = 10.0$ and the uncertainty tolerance $\gamma_{\text{tol}} = 0.4$ MPa fixed and run 10 models with different seeds being given to the pseudo-random number generator used to initialize the BFGS optimizer that finds the hyperparameter values. Results in terms of reduction ratios and load-displacement curves are shown in Fig. 13. With the different initializations, the optimizer finds different local marginal likelihood maxima corresponding to different sets of hyperparameters. This in turn leads to a large spread in acceleration levels between models, once again demonstrating the sensitivity of the framework to the hyperparameter values and to their interaction with γ_{tol}. Nevertheless, all 10 models approximate the reference response with excellent accuracy due to the fairly strict value adopted for γ_{tol}.

5.2.5. Acceleration versus mesh density

We now return to the two-dimensional model shown in Fig. 7 in order to investigate how model performance scales with the level of mesh discretization. We solve the model with multiple different mesh densities with characteristic element sizes ranging from 16 mm (32-element mesh) to 0.8 mm (3020-element mesh). All meshes are composed of constant-strain triangles with one integration point each. The model parameters are the same as in the previous examples and the uncertainty threshold is fixed at $\gamma_{\text{tol}} = 1.0$ MPa. Furthermore, and in order to keep the comparison between meshes as consistent as possible, we first use the 3020-element model to estimate the hyperparameters once at the beginning of the study (by loading monotonically along the k-means direction), set them as initial values for all other meshes and keep them fixed by adopting a high value for L_{retrain}. This avoids the possibility of GP surrogates from models with different meshes converging to different local maxima of the likelihood function with distinct behaviors (see Fig. 13) due to small fluctuations in stress values.

The relative changes in $|\mathcal{D}|$ and $|\mathcal{A}|$ as well as in the average force error with respect to the reference solution are plotted against the discretization level in Fig. 14. It can be seen that the amount of information needed by the model in order to maintain accuracy is independent of the mesh density, with both $|\mathcal{D}|$ and $|\mathcal{A}|$ showing only relatively minor fluctuations as the mesh is refined. This is a consequence of the greedy approach employed here: even though the total number of integration points can be large, points are only sampled or added to $\mathcal{A}$ if their uncertainty is higher than γ_{tol}.

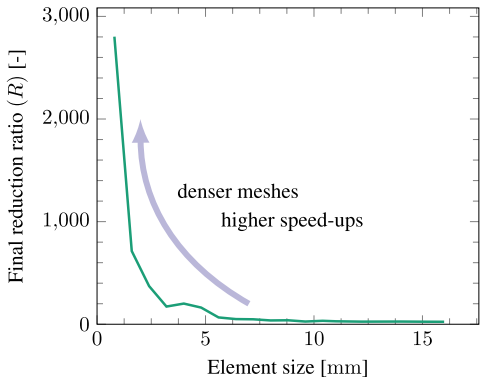


Fig. 15. Final reduction ratios for the two-dimensional bar problem with different levels of mesh discretization. The efficiency of the reduction framework increases significantly as denser meshes are used.

Since models with different meshes are approximating the same underlying solution, it is intuitive to expect the sampling effort to be similar: even though the constitutive manifold is more densely evaluated if a denser mesh is used, these evaluations consist of closely-packed clusters in strain space whose response can be accurately approximated by a small number of GP observations.[7] As a consequence, the reduction ratio R (and therefore the speed-up) increases dramatically with mesh density, as can be seen in Fig. 15. Recalling that the same example has been used with FE^2 in Section 5.1, we can therefore expect that opting for the densest mesh used here would lead to a reduced model almost 3000 times faster than its full-order counterpart without resorting to *offline* training.

5.2.6. Initial number of fully-solved points

As one final parametric study on the two-dimensional model of Fig. 7, we investigate the effect of changing the clustering parameter k that determines the initial number of points in $\mathcal{A}$. Fig. 16 shows a set of heatmaps plotting the total number of times each anchor point is sampled during the analysis for three different values of k. For $k = 1$, we start with a point midway between the load application face and the center of the bar and the greedy data selection approach promptly locates the point at the center of the bar undergoing the largest strains. Additional points are eventually added due to a number of canceled steps, but the model nevertheless concentrates most of the sampling effort at the center and the remaining points remain dormant for the rest of the analysis. The same happens for the models with $k = 5$ and $k = 10$: the model starts with k anchor models that remain dormant and immediately adds another one at the center of the bar from where most of the training data is obtained. Due to the relatively simple strain path of this specific example, increasing k does not seem to be beneficial. However, the greedy framework is able to naturally disregard the redundant information and concentrate the sampling effort where it is needed. All three models have, therefore, similar accuracy and acceleration levels.

5.3. Two-dimensional plate with multiple cutouts

We now move to an example with complex geometry in order to investigate how the reduction framework fares in approximating a larger portion of the original full-order constitutive manifold. The example employs the same elastoplastic material as before but now concerns the plate with multiple cutouts with boundary conditions and final plastic strain distribution shown on Fig. 17. Due to the presence of the cutouts, the stress distribution is considerably more complex than for the previous examples. As the load increases, plastic strain arises at the stress concentration regions between cutouts and forms a strain localization band spanning the complete height of the model. The plate is discretized with 754 constant-strain triangles with one integration point each.

We solve the problem with $k = 10$, $\gamma_{\text{tol}} = 2.0$ MPa, $L_{\text{retrain}} = 10$ and $\gamma_{\text{cancel}} = 80$ MPa for a total of 100 time steps. As in the previous examples, the initial hyperparameters are estimated by sampling the initial k-means directions. We plot the evolution of the reduction ratio R and the size of the fully-solved set $\mathcal{A}$ in Fig. 18. While the model of Fig. 16 is able to rely on the information coming from a single anchor model to accurately describe its constrained constitutive space, the complex stress distribution of the current example demands the sampling of a significantly higher number of points. The acceleration is therefore smaller than for the previous cases, with a reduction ratio of approximately 63 at the end of the analysis. This result is not unexpected, since the reduction framework relies on the assumption that the macroscopic geometry and boundary conditions constrain the constitutive response to lie on a manifold of much lower complexity. The complex stress state treated here challenges this assumption.

[7] This is a natural consequence of our choice of kernel, which quantifies the similarity between points by their distance in strain space.

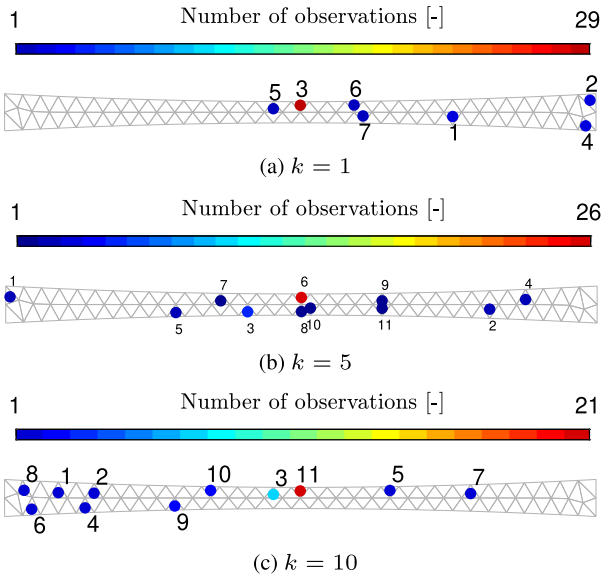


Fig. 16. Heatmaps of the total number of samplings of points in $\mathcal{A}$ for different values of k. The numbers next to the points indicate the order at which the points are added to $\mathcal{A}$. For each case, the first k points are initially present.

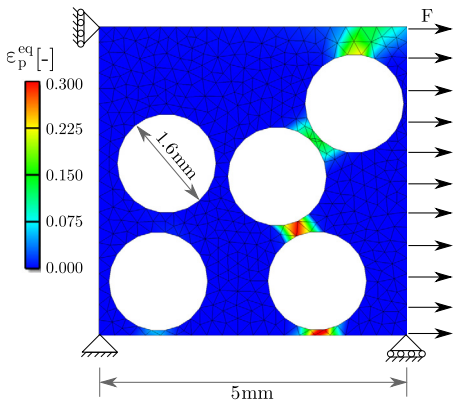


Fig. 17. Geometry, boundary conditions and final equivalent plastic strain distribution for the two-dimensional cutout example.

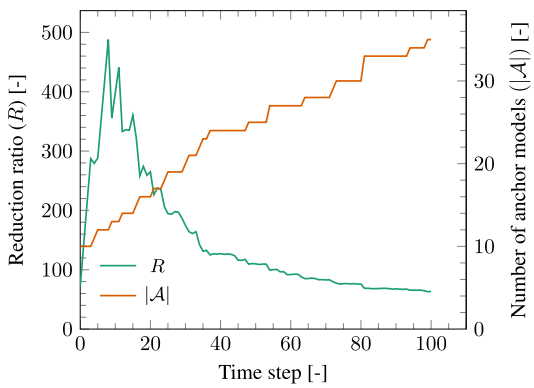


Fig. 18. Evolution of the reduced-order solution of the two-dimensional plate problem with cutouts. With a more complex constitutive behavior to approximate, the size of the $\mathcal{A}$ set is larger for this example.

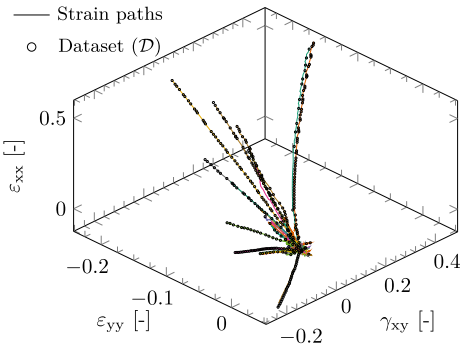


Fig. 19. Strain paths seen by each of the 754 integration points of the reduced model, plotted together with the observations stored in $\mathcal{D}$. The complex geometry of the example leads to a large number of distinct strain paths that must be learned by the GP models.

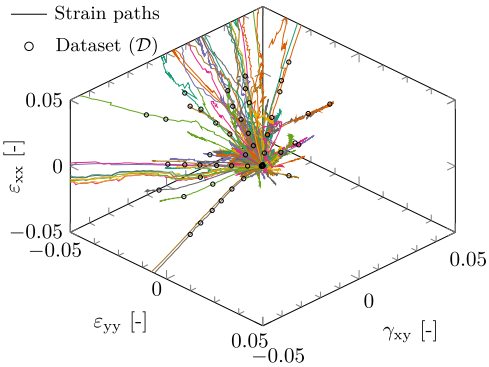


Fig. 20. Zoomed-in view of the strain paths of Fig. 19. Since most points do not undergo strain localization, they can be readily represented by the trained GP models without extensive sampling.

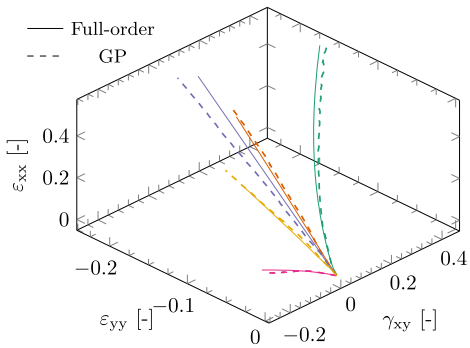


Fig. 21. Strain paths associated with the five integration points with the largest strain norm at the end of the analysis. Paths resulting from the full-order and reduced models are plotted.

This complexity can be visualized by plotting the strain paths seen by different integration points of the reduced model together with the dataset $\mathcal{D}$ in Fig. 19. In contrast with previous examples, on which every point on the mesh follows approximately the same path, here we different points moving towards distinct regions of the parameter space. Looking at the sampling points (marked as black circles in Fig. 19), we see that the framework automatically assigns anchor models to points with diverging paths. For the remaining points, whose strain paths have a higher degree of similarity, the GP surrogates are capable of taking over without extensive sampling, as can be seen in the zoomed region shown in Fig. 20.

Comparing the paths seen by individual points of the full-order and reduced models, the behavior shown in Fig. 21 for the five integration points with the highest final strain norm is obtained. Since the strain path is ultimately determined by the global displacement field, which in turn depends on the performance of the surrogate models, the full-order and reduced paths do not exactly agree, although we still obtain a fairly accurate response.

It is also interesting to plot the heatmap of GP observations in order to visualize which points are being sampled. Results can be seen in Fig. 22. In contrast with the model of Fig. 16, sampling is performed on a larger number of points. This

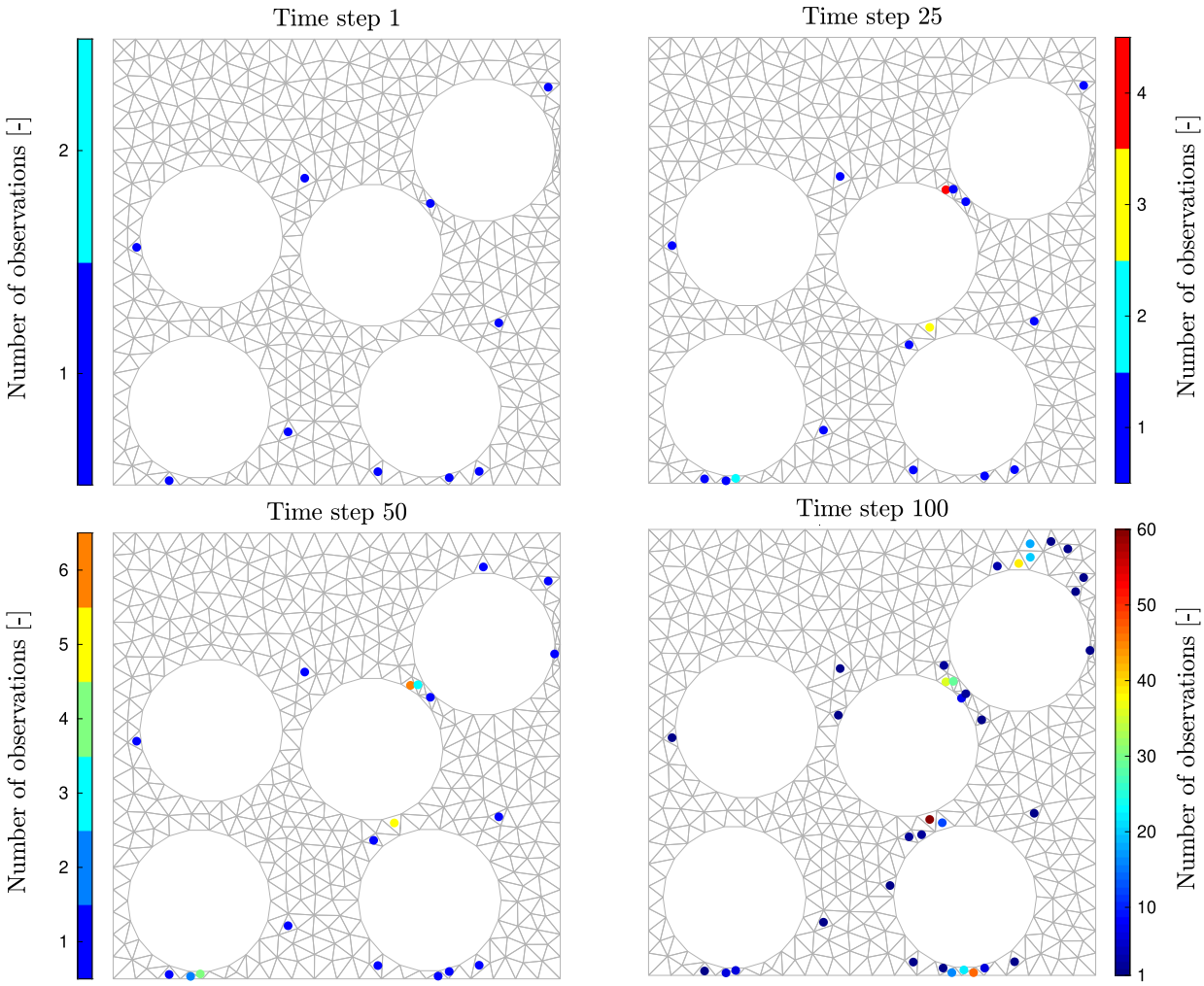


Fig. 22. Heatmaps of the total number of samplings of points in $\mathcal{A}$ for the two-dimensional cutout example taken at four different moments throughout the analysis. The adaptive model automatically concentrates the sampling effort at regions undergoing strain localization.

indicates that different parts of the mesh experience significantly different strain paths. Comparing Fig. 22 with the plastic strain field of Fig. 17, we see that the point distribution is closely related to how plastic strain is distributed throughout the domain. The framework is therefore able to direct computational effort to regions in the mesh where it is most needed while employing an efficient approximation for the rest of the domain.

The fact that most initial anchors in Fig. 22 are never sampled after the first step suggests the model of Fig. 17 is also rather insensitive to the choice of clustering parameter k. However, even though the sampling strategy adaptively adds new anchors where needed and lets anchors with redundant information remain dormant, the value of k affects the initial estimates for the hyperparameters, as can be seen in Fig. 23. Since the present example features a richer variety of strain paths, the initial estimation for the hyperparameters benefits from information coming from a larger set of fictitious anchors. Finally, the load-displacement curves of the full-order and reduced models are shown in Fig. 24. Again a satisfactory agreement is obtained at a fraction of the number of full model evaluations.

It is worth mentioning, however, that this more complex example is less numerically stable than the previous ones. Running the model with higher values of γ_{tol} leads to convergence issues as new data coming from the anchor models leads to large jumps between equilibrium solutions that push the stability of the Newton-Raphson solver to its limit. Additionally, a model with reasonable stability could only be obtained by forcibly increasing the smoothness of the GP approximation by increasing the lower bound of σ_{n}^2 to 1.0 MPa^2. Further research effort is therefore necessary in order to improve the stability of the present approach when faced with highly-complex strain distributions.[8]

[8] Adopting prior distributions for σ_{f}^2, ℓ and σ_{n}^2 might also help in regularizing the GP response and improving solver stability. This merits further investigation.

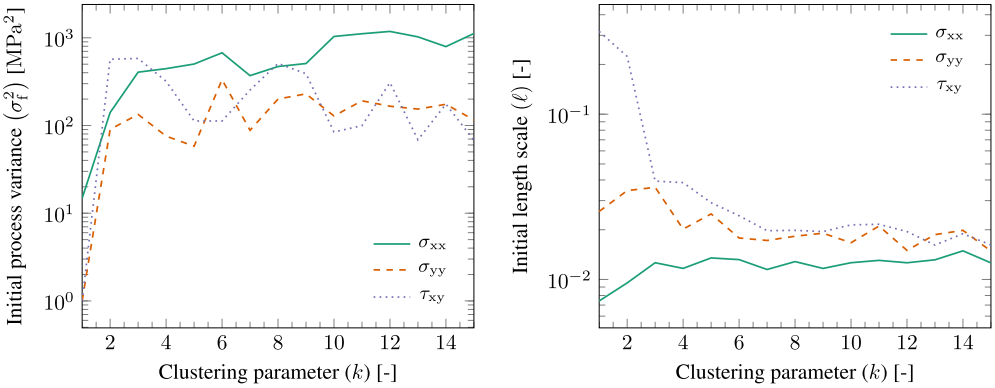


Fig. 23. Initial values for the variance and length scale hyperparameters of the GP models for different values of the clustering parameter k. Although the hyperparameters are adaptively re-estimated during the analysis, different initial values can affect the overall performance of the reduced model.

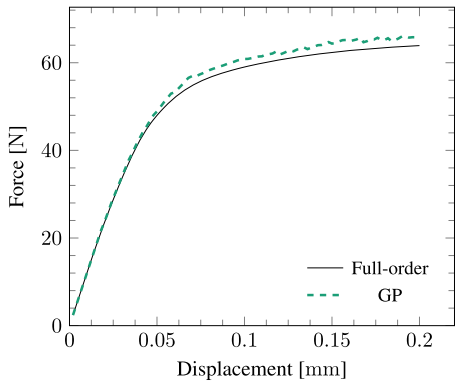


Fig. 24. Load-displacement curves for the two-dimensional plate with cutouts obtained with the full- and reduced-order models.

5.4. Mixed-mode cohesive crack propagation

We close the present discussion with one final example exploring the use of the framework to approximate a traction-separation response. In Section 2.4, we argue that bulk homogenization in FE^2 loses objectivity upon global softening at the microscale, at which point switching to a cohesive homogenization strategy becomes necessary. Here we construct a surrogate model for the associated macroscopic cohesive material by using the framework of Section 4 but now defining $\mathcal{S}$ as:

$$\boldsymbol{\tau}^{\Omega}\left(\boldsymbol{\tau}^{\Omega}_{\text{eff}}, \mathcal{D}\right) = \mathbb{E}\left[\widehat{\boldsymbol{\tau}}\left(\boldsymbol{\tau}^{\Omega}_{\text{eff}}, \mathcal{D}\right)\right] \tag{43}$$

where $\boldsymbol{\tau}^{\Omega}_{\text{eff}}$ is an effective traction computed from the displacement jump $[\![\mathbf{u}]\!]$ and the stress at crack initiation that accounts for the singular nature of the initially-rigid cohesive law and takes the place of the shifted jump $[\![\mathbf{v}]\!]^{\Omega}$ of Eq. (7) as input variable [49]. Note that, in contrast to Eq. (37), assuming a correction from elasticity ceases to be interesting here and we therefore build a direct regression for the tractions with one GP model for each component of the surrogate traction vector $\boldsymbol{\tau}^{\Omega}$. Furthermore, we exploit the knowledge that decohesion is an irreversible process by switching to the trivial solution $\boldsymbol{\tau} = \mathbf{0}$ after the GP confidently predicts zero traction for the first time:

$$\text{for a given point } p\text{, if } \|\mathbb{E}\left[\widehat{\boldsymbol{\tau}}\right]\|_p \leq \sigma_{\text{n}} \text{ and } \gamma^{\text{o}}_p \leq \gamma_{\text{tol}} \;\Rightarrow\; \text{switch to } \boldsymbol{\tau}^{\Omega} = \mathbf{0} \,|\, \forall\, \boldsymbol{\tau}^{\Omega}_{\text{eff}} > \mathbf{0} \tag{44}$$

After the switch, we stop using the GP approximation for the integration point in question. This modification improves efficiency because we avoid having to train the GP to reproduce the fully-damaged branch of the cohesive law.[9]

The example concerns the mixed-mode bending test shown in Fig. 25 and is taken from [49], where the same structure is solved for multiple mode-mixity ratios. Here we opt for a single ratio $\alpha = G_{\text{II}}/\left(G_{\text{I}} + G_{\text{II}}\right) = 0.5$ and therefore only deal

[9] The GP prediction naturally moves to its zero-traction prior away from $\mathcal{D}$, but retraining would still be periodically triggered due to the variance increase.

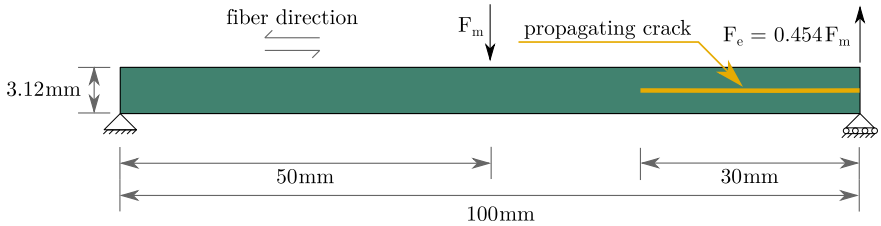


Fig. 25. Mixed-mode crack propagation example: Geometry, loads and boundary conditions.

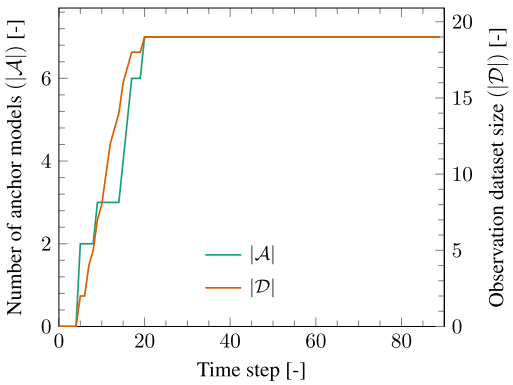


Fig. 26. Mixed-mode crack propagation example: Evolution of the learning process in terms of the sizes of the $\mathcal{A}$ and $\mathcal{D}$ sets. No learning occurs during most of the crack propagation process. The framework also refrains from computing the full-order cohesive model for most of the analysis.

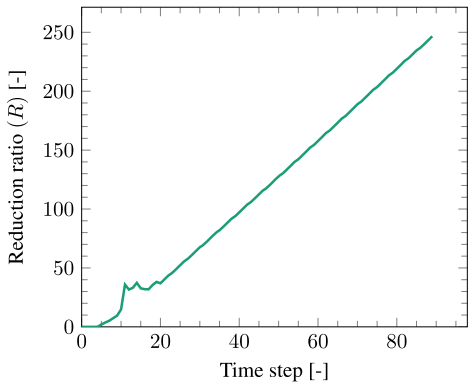


Fig. 27. Evolution of the reduction ratio R for the mixed-mode crack propagation example. The fact that $\mathcal{D}$ is not growing for most of the analysis allows for high acceleration ratios to be obtained.

with the specific ratio between applied forces shown in Fig. 25 and with a single initial notch length of 30 mm. In an FE^2 approach, both the bulk material behavior and the cohesive softening response would be derived from embedded RVEs by employing Eqs. (5) and (7). Without loss of generality and in keeping with the original model of [49], in this demonstration we instead use a linear-elastic orthotropic model for the bulk material and a bulk stress-constrained cohesive zone law to model the traction-separation behavior of the propagating crack. The model is solved in plane stress and initially discretized with 3107 4-node quadrilateral elements with 4 integration points each. Cohesive segments are inserted on the fly by using the Phantom Node method [50] (later renamed CutFEM [51]), with elements being duplicated in order to describe a displacement jump running through the elements as the crack propagates from the tip of the notch.

In order to keep the discussion simple, we only use the GP framework to approximate the response at cohesive integration points, but extending the example to also use GP for the bulk response would be straightforward. We run the reduced model with $\gamma_{\text{tol}} = 1.0$ MPa, $\gamma_{\text{cancel}} = 100$ MPa and fixed hyperparameters obtained from sampling a fictitious anchor model in the initial effective traction direction seen by the first cohesive point. Note that in this case we cannot rely on the k-means strategy from the previous examples since the analysis starts with no cohesive integration points.

The solution process is tracked by plotting the evolution of the $\mathcal{A}$ and $\mathcal{D}$ sets on Fig. 26. During the first five time steps, the structure is loaded until the onset of crack propagation. As no cohesive points exist at this stage, $\mathcal{A}$ and $\mathcal{D}$ remain empty. The first cohesive integration points created when the crack starts to propagate are added to $\mathcal{A}$ and their responses

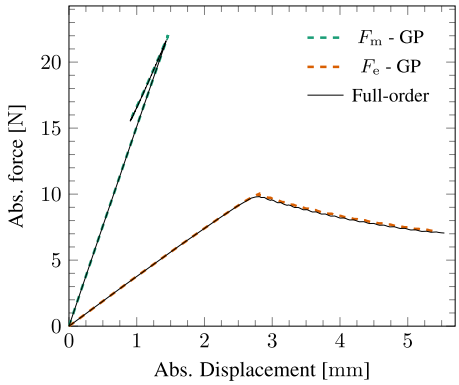


Fig. 28. Load-displacement curves for the mixed-mode crack propagation example. The active learning framework shows excellent agreement with the full-order response.

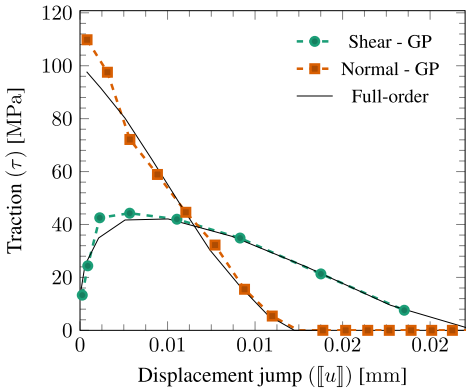


Fig. 29. Evolution of the mixed-mode decohesion of an integration point computed by the GP model during the analysis phase on which $\mathcal{D}$ is not growing.

are sampled into $\mathcal{D}$. From that moment on, the framework is left to decide which points are added. Since the crack tip can be seen as a moving source travelling through the domain, points created after time step 20 can be accurately approximated with information obtained from points closer to the notch. Since the decohesion process follows almost exactly the same path in these subsequent integration points and we switch to a trivial solution with zero traction for fully-damaged points (Eq. (44)), no new data is needed in $\mathcal{D}$ for the rest of the analysis. The greedy algorithm is able to detect this and interrupt the learning process, with the 7 anchor models in $\mathcal{A}$ remaining dormant for the rest of the analysis and no more full-order updates being computed. This results in the high values for the reduction ratio plotted in Fig. 27.

The load-displacement curves obtained with the full and hybrid models are shown in Fig. 28, where we plot absolute forces and displacements from the two load locations shown in Fig. 25. It can be seen that the global response obtained with the active learning approach is virtually indistinguishable from the full-order one. Finally, we can observe how accurate the local traction approximation is by plotting in Fig. 29 the traction-separation curves for a cohesive integration point created and completely solved after the anchor models become dormant. We see that the complete mixed-mode softening process is correctly predicted by the GP models based solely on the response of earlier points.

6. Conclusions

This work introduces an adaptive probabilistic learning framework for the *online* construction of surrogate constitutive models for concurrent multiscale analysis. The framework eliminates the need to sample a potentially infinitely-dimensional input space *offline* and instead fits a set of Gaussian Process (GP) models with data sampled *online* from a small number of fully-solved *anchor* models. The approach incorporates additional physics information by enhancing the conventional GP regression with tangent stiffness observations coming from the anchor models. A greedy data selection procedure ensures the surrogate response is kept accurate, efficient and independent of the time step size and of the macroscopic discretization level.

The reduction approach was described in detail and its performance was assessed with an extensive set of numerical examples. The ability of the framework of reducing the computational effort associated with FE^2 was demonstrated with a preliminary example, after which a detailed parametric study was performed on single-scale models without loss of generality. The uncertainty tolerance parameter used to control the GP sampling frequency was found to provide only a

limited degree of control on the balance between accuracy and efficiency of the reduced response due to the presence of other model components that can also trigger a refinement of the GP approximations (*e.g.* canceled solutions). Using a larger initial set of fully-solved points or allowing for the GP hyperparameters to be re-estimated during the analysis were found to exert little influence on the performance of the model, at least for the specific examples treated here. The acceleration brought by the greedy learning strategy was found to drastically increase as the macroscopic mesh is refined, with reduction ratios as high as 2800 times being obtained.

An additional example was used to demonstrate the ability of the reduction approach to handle models with complex stress distributions. Although the acceleration was lower in this case due to the complexity of the constitutive manifold being approximated, the greedy sampling strategy successfully concentrated the learning effort on the most informative mesh regions. One final model involving mixed-mode crack propagation was presented. In contrast with the previous examples, the nature of the crack propagation problem allowed the GP to reuse previously obtained information rather than continuously growing the dataset. Acceleration ratios of up to 250 times were obtained and the GP surrogate was able to take over the entire set of integration points from the original full-order model for most of the analysis.

The presented results suggest the framework is a promising approach in reducing the computational effort of nonlinear concurrent multiscale modeling and circumventing the curse of dimensionality associated with the *offline* construction of surrogates for path-dependent materials. Nevertheless, further model development is necessary in order to allow for non-monotonic load paths including unloading/reloading behavior since the adopted GP formulation assumes a unique mapping between strains and stresses. Although path dependency is accounted for in the sense that macroscopic models with different strain paths will construct different surrogates, an accurate approximation is not guaranteed for models that visit points in strain space more than once during their loading paths.

Data availability

The datasets collected *online* and used to train the GP models for all examples in this article are available at the 4TU.ResearchData repository through https://doi.org/10.4121/13019864.

CRediT authorship contribution statement

I.B.C.M. Rocha: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **P. Kerfriden:** Conceptualization, Methodology, Supervision, Writing – review & editing. **F.P. van der Meer:** Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing.

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.

Acknowledgements

The authors gratefully acknowledge financial support from the Netherlands Organisation for Scientific Research (NWO) under Vidi grant nr. 16464.

References

[1] S. Gantenbein, K. Masania, W. Woigk, J.P.W. Sesseg, T.A. Tervoort, A.R. Studart, Three-dimensional printing of hierarchical liquid-crystal-polymer structures, Nature 561 (2018) 226–230.
[2] F. Feyel, Multiscale fe^2 elastoviscoplastic analysis of composite structures, Comput. Mater. Sci. 16 (1999) 344–354.
[3] M.G.D. Geers, V.G. Kouznetsova, W.A.M. Brekelmans, Multi-scale computational homogenization: trends and challenges, J. Comput. Appl. Math. 234 (2010) 2175–2182.
[4] V. Kouznetsova, W.A.M. Brekelmans, F.P.T. Baaijens, An approach to micro-macro modeling of heterogeneous materials, Comput. Mech. 27 (2001) 37–48.
[5] V.P. Nguyen, O. Lloberas-Valls, M. Stroeven, L.J. Sluys, Computational homogenization for multiscale crack modelling. Implementation and computational aspects, Int. J. Numer. Methods Eng. 89 (2012) 192–226.
[6] I. Özdemir, W.A.M. Brekelmans, M.G.D. Geers, Computational homogenization for heat conduction in heterogeneous solids, Int. J. Numer. Methods Eng. 73 (2008) 185–204.
[7] K. Terada, M. Kurumatani, Two-scale diffusion-deformation coupling model for material deterioration involving micro-crack propagation, Int. J. Numer. Methods Eng. 83 (2010) 426–451.
[8] I.B.C.M. Rocha, F.P. van der Meer, R.P.L. Nijssen, L.J. Sluys, A multiscale and multiphysics numerical framework for modelling of hygrothermal ageing in laminated composites, Int. J. Numer. Methods Eng. 112 (2017) 360–379.
[9] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2002) 492–515.
[10] P. Kerfriden, P. Gosselet, S. Adhikari, S.P.A. Bordas, Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems, Comput. Methods Appl. Math. 200 (2011) 850–866.
[11] J.A. Hernández, M.A. Caicedo, A. Ferrer, Dimensional hyper-reduction of nonlinear finite element models via empirical cubature, Comput. Methods Appl. Math. 313 (2017) 687–722.

[12] R.A. van Tuijl, C. Harnish, K. Matouš, J.J.C. Remmers, M.G.D. Geers, Wavelet based reduced order models for microstructural analyses, Comput. Mech. 63 (2019) 535–554.
[13] Z. Liu, M. Bessa, W.K. Liu, Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials, Comput. Methods Appl. Math. 306 (2016) 319–341.
[14] O. Goury, D. Amsallem, S.P.A. Bordas, W.K. Liu, P. Kerfriden, Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization, Comput. Mech. 58 (2016) 213–234.
[15] I.B.C.M. Rocha, F.P. van der Meer, L.J. Sluys, Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling, Comput. Methods Appl. Math. 345 (2019) 644–670.
[16] B.A. Le, J. Yvonnet, Q.C. He, Computational homogenization of nonlinear elastic materials using neural networks, Int. J. Numer. Methods Eng. 104 (2015) 1061–1084.
[17] X. Lu, D.G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, J. Bai, A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites, Comput. Mech. 64 (2) (2018) 307–321.
[18] F. Ghavamian, A. Simone, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, Comput. Methods Appl. Math. 357 (2019) 23p.
[19] Z. Liu, C.T. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, Comput. Methods Appl. Math. 345 (2019) 1138–1168.
[20] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, Deep learning predicts path-dependent plasticity, Proc. Natl. Acad. Sci. 116 (2019) 26414–26420.
[21] I.B.C.M. Rocha, P. Kerfriden, F.P. van der Meer, Micromechanics-based surrogate models for the response of composites: a critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks, Eur. J. Mech. A, Solids 82 (2020) 103995.
[22] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D.W. Apley, C. Brinson, W. Chen, W.K. Liu, A framework for data-driven analysis of materials under uncertainty: countering the curse of dimensionality, Comput. Methods Appl. Math. 320 (2017) 633–667.
[23] P. Kerfriden, O. Goury, T. Rabczuk, S.P.A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, Comput. Methods Appl. Math. 256 (2013) 169–188.
[24] F. Ghavamian, P. Tiso, A. Simone, POD-DEIM model order reduction for strain-softening viscoplasticity, Comput. Methods Appl. Math. 317 (2017) 458–479.
[25] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, J. Comput. Phys. 202 (2005) 346–366.
[26] I.B.C.M. Rocha, F.P. van der Meer, L.J. Sluys, An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training, Comput. Methods Appl. Math. 358 (2020) 112650.
[27] F. Fritzen, M. Fernández, F. Larsson, On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling, Front. Mater. 6 (2019) 75.
[28] A.M. Tartakovsky, D.A. Barajas-Solano, Machine learning for quantifying and reducing model uncertainty in non-linear partial differential equation models, in: AGU Fall Meeting Abstracts, 2018.
[29] M. Salloum, J.A. Templeton, Inference and uncertainty propagation of atomistically informed continuum constitutive laws, part 2: generalized continuum models based on Gaussian processes, Int. J. Uncertain. Quantificat. 4 (2014) 171–184.
[30] A. Frankel, R. Jones, L. Swiler, Tensor basis Gaussian process models of hyperelastic materials, arXiv:1912.10872, 2019.
[31] J. Knap, N.R. Barton, R.D. Hornung, A. Arsenlis, R. Becker, D.R. Jefferson, Adaptive sampling in hierarchical simulation, Int. J. Numer. Methods Eng. 76 (2008) 572–600.
[32] N.R. Barton, J. Knap, A. Arsenlis, R. Becker, R.D. Hornung, D.R. Jefferson, Embedded polycrystal plasticity and adaptive sampling, Int. J. Plast. 24 (2008) 242–266.
[33] C.L.C. Mattos, Z. Dai, A. Damianou, J. Forth, G.A. Barreto, N.D. Lawrence, Recurrent Gaussian processes, arXiv:e-prints, 2016, arXiv:1511.06644v6.
[34] A. Doerr, C. Daniel, M. Schiegg, D. Nguyen-Tuong, S. Schaal, M. Toussaint, S. Trimpe, Probabilistic recurrent state-space models, arXiv:e-prints, 2018, arXiv:1801.10395v2.
[35] R. Hill, A self-consistent mechanics of composite materials, J. Mech. Phys. Solids 12 (1965) 213–222.
[36] V.P. Nguyen, O. Lloberas Valls, M. Stroeven, L.J. Sluys, On the existence of representative volumes for softening quasi-brittle materials - a failure zone averaging scheme, Comput. Methods Appl. Math. 199 (2010) 45–48.
[37] P.J. Sánchez, P.J. Blanco, A.E. Huespe, R.A. Feijóo, Failure-oriented multi-scale variational formulation: micro-structures with nucleation and evolution of softening bands, Comput. Methods Appl. Math. 257 (2013) 221–247.
[38] J. Oliver, M. Caicedo, E. Roubin, A.E. Huespe, J.A. Hernández, Continuum approach to computational multiscale modeling of propagating fracture, Comput. Methods Appl. Math. 294 (2015) 384–427.
[39] E. Svenning, F. Larsson, M. Fagerström, A two-scale modeling framework for strain localization in solids: XFEM procedures and computational aspects, Comput. Struct. 211 (2019) 43–54.
[40] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[41] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT, 2006.
[42] E. Solak, R. Murray-smith, W.E. Leithead, D.J. Leith, C.E. Rasmussen, Derivative observations in Gaussian process models of dynamic systems, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, 2003, pp. 1057–1064.
[43] J. Prüher, S. Särkkä, On the use of gradient information in Gaussian process quadratures, in: Proceedings of the 2016 IEEE International Workshop on Machine Learning for Signal Processing, 2016.
[44] R. Fletcher, Practical Methods of Optimization, Wiley, 1987.
[45] T.C. Germann, Adaptive physics refinement at the microstructure scale, in: W. Andreoni, S. Yip (Eds.), Handbook of Materials Modeling, Springer, 2020.
[46] S.P. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theory 28 (1982) 129–137.
[47] Jive - Software development kit for advanced numerical simulations, http://jive.dynaflow.com. (Accessed 8 November 2019).
[48] A.R. Melro, P.P. Camanho, F.M. Andrade Pires, S.T. Pinho, Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I - constitutive modelling, Int. J. Solids Struct. 50 (2013) 1897–1905.
[49] F.P. Van der Meer, L.J. Sluys, A phantom node formulation with mixed mode cohesive law for splitting in laminates, Int. J. Fract. 158 (2009) 107–124.
[50] A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, Comput. Methods Appl. Math. 193 (2004) 3523–3540.
[51] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, Cutfem: discretizing geometry and partial differential equations, Int. J. Numer. Methods Eng. 104 (2015) 472–501.