A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality

Article *in* Computer Methods in Applied Mechanics and Engineering · April 2017
DOI: 10.1016/j.cma.2017.03.037

CITATIONS
19

READS
131

8 authors, including:

M. A. Bessa
Delft University of Technology
18 PUBLICATIONS **221** CITATIONS
SEE PROFILE

Ramin Bostanabad
Northwestern University
10 PUBLICATIONS **63** CITATIONS
SEE PROFILE

Zeliang Liu
LSTC
14 PUBLICATIONS **68** CITATIONS
SEE PROFILE

Wing Kam Liu
Northwestern University
390 PUBLICATIONS **22,054** CITATIONS
SEE PROFILE

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 320 (2017) 633–667
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality

M.A. Bessa[a,b,1], R. Bostanabad[b], Z. Liu[d], A. Hu[b], Daniel W. Apley[c], C. Brinson[b], W. Chen[b], Wing Kam Liu[b,*]

[a] *California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125, USA*[2]
[b] *Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA*
[c] *Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA*
[d] *Theoretical and Applied Mechanics, Northwestern University, Evanston, IL 60208, USA*

Received 1 November 2016; received in revised form 17 February 2017; accepted 24 March 2017
Available online 1 April 2017

Highlights

- Unified data-driven framework for design and modeling of materials and structures.
- Integration of design of experiments, computational analyses, and machine learning.
- Avoids curse of dimensionality when using self-consistent clustering analyses method.

Abstract

A new data-driven computational framework is developed to assist in the design and modeling of new material systems and structures. The proposed framework integrates three general steps: (1) design of experiments, where the input variables describing material geometry (microstructure), phase properties and external conditions are sampled; (2) efficient computational analyses of each design sample, leading to the creation of a material response database; and (3) machine learning applied to this database to obtain a new design or response model.

In addition, the authors address the longstanding challenge of developing a data-driven approach applicable to problems that involve unacceptable computational expense when solved by standard analysis methods – e.g. finite element analysis of representative volume elements involving plasticity and damage. In these cases the framework includes the recently developed "self-consistent clustering analysis" method in order to build large databases suitable for machine learning. The authors believe

* Corresponding author.
E-mail address: w-liu@northwestern.edu (W.K. Liu).
[1] Contact M.A. Bessa (mbessa@caltech.edu) if interested in the data-driven framework code.
[2] Current address.

http://dx.doi.org/10.1016/j.cma.2017.03.037

that this will open new avenues to finding innovative materials with new capabilities in an era of high-throughput computing ("big-data").

Keywords: Design of experiments; Reduced order model; Self-consistent clustering analysis; Machine learning and data mining; Plasticity

1. Introduction

Structural and materials design is a highly iterative process where one seeks an optimal design for chosen quantities of interest. Even the simplest structures and materials are composed by multiple building blocks that can be combined in a large number of possibilities. These building blocks together with the range of boundary conditions applied to the material/structure can lead to drastically different optimal designs.

For the particular case of material systems design, the high-dimensionality of the engineering design space is striking when considering the overwhelming amount of possible combinations that lead to different materials. Meyers et al. [1] examined the extraordinary diversity in biological materials that are often composed of weak constituents assembled in complex structures with radically different macroscopic mechanical properties. Jang et al. [2] designed three-dimensional hollow ceramic nanostructures inspired by the observation of biological structures with hierarchical arrangements of basic structural elements. Wang et al. [3] explored the design of elastic beam elements attached to an elastomeric core matrix to obtain a metamaterial with intriguing acoustic properties. Other examples can be encountered in investigations on the influence of nano- and micro-reinforcements on the behavior of elastomers [4–6], polymeric foams [7], metal matrix composites [8], and the effect of fiber hybridization in polymer matrix composites [9,10].

This large dimension of the engineering design space represents a major obstacle to accelerate the design process. There are simply too many possibilities to conduct experimental investigations for every conceptual design. Therefore, different authors have proposed to explore the design space via data-driven computational analyses [11–15]. Multiple data-driven methodologies have been presented in the literature. Kirchdoerfer and Ortiz [15] developed a variational approach and applied it to linear elastic truss structures, where the goal was to pair specific data points to local material states such that essential constraints and conservation laws were satisfied while minimizing the distance between these states and the associated data selections. Other methodologies are based on optimization algorithms that try to adaptively find optimal designs or quantities of interest – see appropriate literature in topology optimization [16–18] and other optimization problems [19–21] often using genetic algorithms that tend to find local optima when problems are sufficiently nonlinear and complex.

The data-driven computational framework presented herein is developed for machine learning and often requires large databases. The advantage is that the complete relationship between the key descriptors of each design and the quantities of interest is approximated, enabling its use for distinct purposes. For example, finding general constitutive models for materials as a function of their microstructure and phase properties, or predicting the *global optimum* design for the material within the sampled space.

Fig. 1 illustrates the proposed data-driven framework where three different research fields are integrated: (1) design of experiments (DoE); (2) computational analyses and homogenization; and (3) machine learning (or data mining).

Each of the above steps is a field of research in itself where multiple achievements have been observed in recent years. Therefore, Sections 1.1, 1.2 and 1.3 provide a short review of the major accomplishments in the respective step of the proposed framework. Then, the procedure is applied to solve two distinct problems: (1) finding a hyperelastic composite constitutive law where standard computational analyses can be used to build the database; and (2) maximizing composite toughness by using the self-consistent clustering analysis to accelerate the plasticity and fracture predictions in order to build the necessary database.

1.1. Design of experiments (DoE)

Once the problem and objectives are defined, then the first step is to perform the design of experiments (DoE) for characterizing the high dimensional space of input variables by a finite set of descriptors. As shown in Box 1 of Fig. 1,

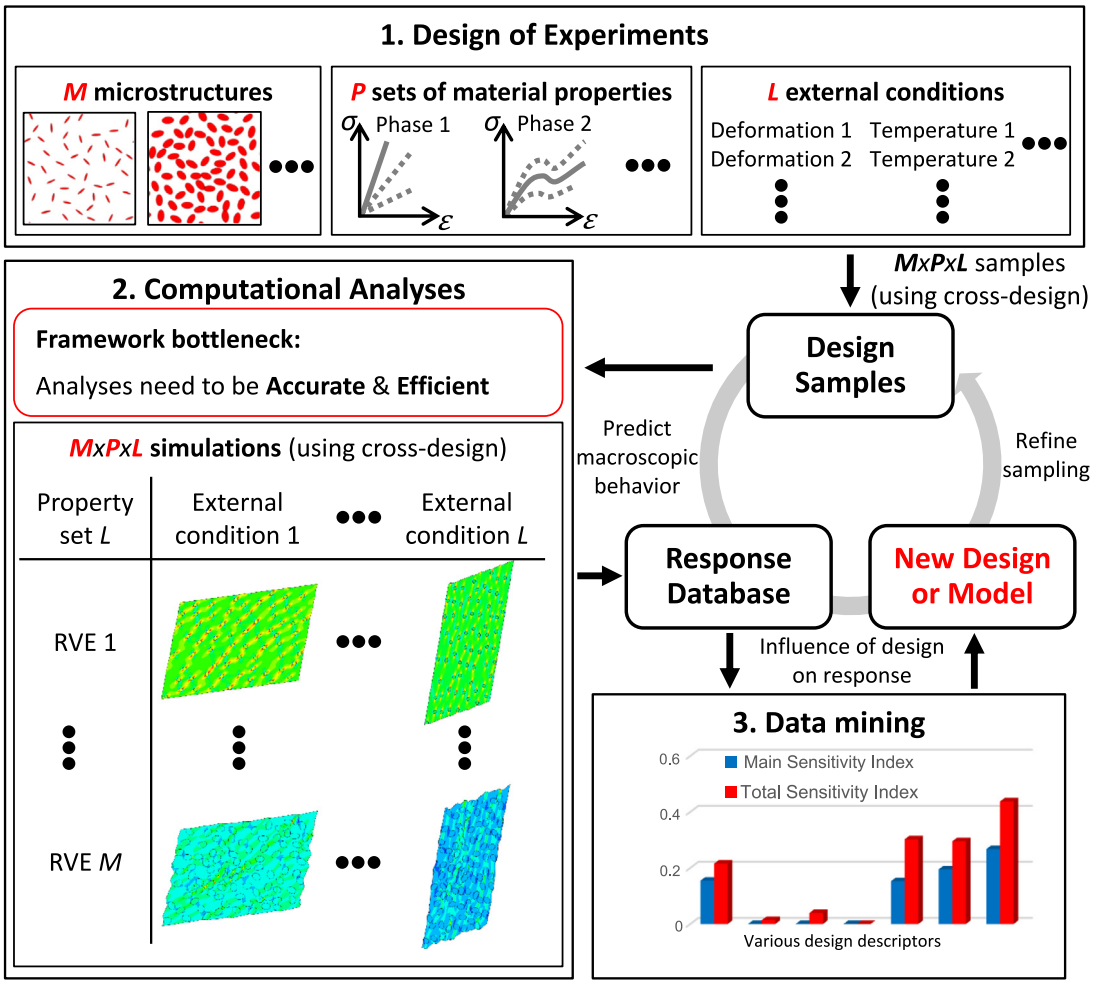


Fig. 1. Schematic of global framework for data-driven material systems design/modeling.

three types of descriptors are identified for the design of material systems: microstructural (geometric) descriptors, material property descriptors, and external (boundary) conditions descriptors.

The above descriptors contain information about the material building blocks and respective properties, as well as their geometric assembly into material microstructures. The first set of such descriptors is identified via microstructure characterization [22–25] and once defined, one can assign specific material properties and a set of boundary conditions, as illustrated in Box 1 of Fig. 1. Each microstructure is characterized by a representative volume element[3] (RVE) or multiple statistical volume elements[4] (SVEs) that are generated by a fixed set of microstructural descriptors.

Once the descriptors are selected they form a group of input design variables $\mathbf{x}$ for the problem,

$$\mathbf{x} = [x_1, \ \ldots \ , x_i, \ \ldots \ , x_{d_{in}}] \tag{1}$$

where d_{in} indicates the total number of input design variables $\mathbf{x}$ considered. These variables include all the above mentioned descriptors. For example, if the identified microstructure descriptors were volume fraction V_f and number

[3] A representative volume element (RVE) is a material domain randomly generated for a fixed set of material descriptors that has the same homogenized behavior independently of the randomization process. In other words, the stochasticity of the microstructure is not reflected on its macroscopic response.

[4] A statistical volume element (SVE) is a material domain that is not sufficiently large to obtain the same macroscopic material response.

of particles N_p, the property descriptors were the Young's modulus of the particles E_p and Poisson ratio ν_p, and the boundary conditions were given by the three applied strain components ε_{11}, ε_{22}, and ε_{12}, then there would be $d_{in} = 7$ input design variables: $x_1 = V_f$, $x_2 = N_p$, $x_3 = E_p$, $x_4 = \nu_p$, $x_5 = \varepsilon_{11}$, $x_6 = \varepsilon_{22}$, and $x_7 = \varepsilon_{12}$.

After identifying the input design variables and their respective bounds and constraints, design of experiments (DoE) can be used to effectively explore the domain. Without assuming prior knowledge of the problem to be solved, space-filling designs [26,27] that treat different regions of the input variables' domain equally are particularly appropriate. This type of DoE ensures that the sample points are spread out so as to maximize the determinant of the information matrix (i.e., to ensure D-optimality) [28].

Santiago et al. [29] reviewed some of the most common space-filling design methods – e.g. low discrepancy sequences [30–33], good lattice points [34], Latin Hypercube Sampling [35,36] and orthogonal Latin Hypercube Sampling [37] – and classified them according to two metrics, the Maximin (maximizing the minimum Euclidean distance) and the cover measure. This comparative analysis [29] reveals that different optimum Latin Hypercube Samplings [35] and the Sobol sequence [32,33] offer a good compromise between a regular grid and a random distribution.

The design of experiments for all the examples considered herein was performed using Sobol sequence [32,33]. Variants of the Latin Hypercube Sampling were also tested without finding significant differences, so they are not reported. Additional details can be found on the above mentioned literature as well as [26,38]. An example of a DoE with Sobol sequence is given later, see Fig. 3.

1.2. Computational analyses and the curse of dimensionality

The second step is to create a database with the quantities of interest $\mathbf{q}$ that represent the macroscopic response of the material for each design point $\mathbf{x}$ in the DoE. This database is the training dataset that is necessary for the subsequent machine learning step.

The database to be created consists of S sets of inputs $\mathbf{x}$ and outputs $\mathbf{q}$,

$$\{(\mathbf{x}^{(1)}, \mathbf{q}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{q}^{(2)}), \ \ldots\ , (\mathbf{x}^{(S)}, \mathbf{q}^{(S)})\} \tag{2}$$

or in index notation,

$$\{(x_j^{(1)}, q_i^{(1)}), (x_j^{(2)}, q_i^{(2)}), \ \ldots\ , (x_j^{(S)}, q_i^{(S)})\} \quad \text{for } j = 1, \ldots, d_{in} \text{ and } i = 1, \ldots, d_{out} \tag{3}$$

where d_{out} corresponds to the number of output quantities of interest $\mathbf{q}$, and d_{in} was introduced previously as the number of input design variables $\mathbf{x}$. An example of quantities of interest $\mathbf{q}$ is given by the average second Piola–Kirchhoff stress components $q_1 = \bar{S}_{11}$, $q_2 = \bar{S}_{22}$, and $q_3 = \bar{S}_{12}$ of two-dimensional RVEs, as in Section 2.

The response database to be obtained for each DoE point cannot typically be created exclusively from experimental analyses because even simple experiments for each design point can take several hours/days/weeks. An exception occurs for cases when a large experimental database is already available as the result of a coordinated effort from several experimentalists over a long period of time.

When such experimental databases are not available, computational analyses become an attractive option if two requirements are satisfied: (1) the predictions are sufficiently accurate, i.e. the analyses have high-fidelity; and (2) the high-fidelity analyses are not computationally expensive. These requirements are conflicting because high-fidelity simulations usually involve complex material models and fine numerical discretizations leading to large computational expense, as highlighted in Remark 1.

Remark 1. Conducting high-fidelity analyses at a low computational cost represents the core challenge in the proposed framework. On the one hand, without accurate predictions the created database does not represent the real behavior of the material/structure and the model or design found by machine learning is not guaranteed to be useful. On the other hand, without efficient analyses of every sampled microstructure, property set and boundary condition, then the response database becomes insufficiently small for machine learning.

Appendix A includes the general algorithm needed to create the response database of material systems[5] by performing computational analyses of each DoE point. Each DoE point is characterized by a given realization r

[5] The framework could be applied to macroscopic structures instead. In this case, the homogenization procedure may not be needed.

of a microstructure m, a set of properties p of the material phases (building blocks), and external loading conditions l. These DoE points are analyzed by the finite element method of the respective RVE, or by other numerical methods such as meshfree [39–41] or isogeometric [42] with the necessary adaptations. Subsequently, the RVE is homogenized in order to obtain the macroscopic response of the material characterized by a set of quantities of interest that are then stored in a database.

In some cases standard computational analyses such as the finite element method are not sufficiently efficient to be directly used in the data-driven framework. Analyses that are inherently complex and require several days to complete in a high performance computer represent an obvious case, e.g. three-dimensional analyses of heterogeneous RVEs under irreversible deformation (plasticity and/or damage) [10,43–45]. Another case occurs even when the computational analyses are not traditionally considered costly (e.g. hyperelasticity) but the dimensionality of the design space is high enough to require too many sampling points to be evaluated in a timely manner through traditional methods – the curse of dimensionality (increasing the amount of input variables leads to an exponential increase of sampling points needed).

The above mentioned cases cause a bottleneck in the database creation step of the framework, as identified in Box 2 of Fig. 1. In the presence of such bottleneck there needs to be a viable way to accelerate the computational predictions without compromising accuracy for each DoE point. This can be achieved by using reduced order models (ROMs) in the data-driven framework, instead of performing direct numerical simulations (DNS).

Conceptually, there is no important change to the data-driven framework if using a ROM instead of performing a DNS – see Appendix A. These are just analysis methods to determine the response of the system for each design point determined from the DoE. In practice, however, finding the adequate ROM that is both efficient and accurate for the physical process of interest can be a daunting task, especially because ROMs need to be trained in an offline stage before they can be predictive.

A recent ROM proposed by Liu et al. [46] called "self-consistent clustering analysis" (SCA) was developed such that minimal time and computational resources are required for the offline sage, while still leading to accurate predictions for irreversible processes such as inelastic deformation of highly heterogeneous materials. An overview of this method is provided in Appendix B, but the reader is referred to the original publication for details [46].

Other ROMs can be used in the data-driven framework. See for example, micromechanics-based methods [47,48], the transformation field analysis (TFA) [49], the nonuniform transformation field analysis (NTFA) [50], the principal component analysis [51–53] also known as proper orthogonal decomposition (POD) [54,55], and the proper generalized decomposition (PGD) [56,57]. Successful applications of the NTFA [58–60], POD [54,55,61,62] and PGD [63] demonstrate the usefulness of these approaches. However, the simplicity of the SCA method [46] and its applicability to highly localized deformation processes make it particularly attractive for integration in the framework presented herein, as demonstrated next.

1.3. Machine learning

Once the response database is complete, a model that captures the influence of the DoE descriptors on the quantities of interest can be constructed. For small databases these models can be obtained by simple calibration procedures or by trial and error. This has been the traditional approach for developing constitutive models for materials that are based on a relatively small number of experiments.

Nevertheless, a data-driven framework often leads to large databases that need to be evaluated with machine learning (or data mining) methods. Machine learning is at the intersection of high-performance computing, statistics and data sharing [64]. This implies a continued development of computer hardware, statistical theories and numerical methods, as well as the creation and maintenance of open databases [65,66].

Machine learning has drawn particular attention in recent years with the advances in high-throughput computing technology that led to an outbreak of "big data" collection in a wide range of scientific fields [67,68]. One of the most successful and widely spread achievements was the mapping of the human genome [69]. Other notable examples, among many possibilities, can be found in the following fields: neuroscience with the unveiling of secrets of the brain [70]; clinical medicine [71] by finding pathways to cure cancer [72–74]; biology with explanations for the presence of microbiota in the human gut [75] and termites [76], protein analysis and genetics [77]; biotechnology with the design of new drugs [78,79]; artificial intelligence with robots adapting like animals [80] and programs mastering the GO game [81]; psychology by profiling neuropsychosocial behavior [82]; climate change and its effects on food [83]; earthquake detection [84]; and economics [85].

Machine learning is also leading to outstanding achievements in the design of materials at different length and time scales. On the one hand, material scientists have focused on accelerating materials design by probing large datasets obtained from first principle calculations [86–88] or even from failed experiments [89]. Initiatives such as AFLOWlib [90] or the Open Quantum Materials Database [91] are reshaping materials design at the quantum-scale. On the other hand, the pioneering work of Yvonnet and co-workers [11–14] recently opened new avenues for the design of materials at larger scales (micro to macro) by performing the data-driven computational homogenization of nonlinear elastic composite RVEs.

The recent work of Yvonnet and co-workers [12–14] is of particular importance to this article because it demonstrated for the first time that a data-driven framework can be used to determine the constitutive law of nonlinear elastic heterogeneous materials. In principle, their framework can be extended to nonlinear irreversible processes such as plasticity and damage, if the computational cost associated with the analyses of such RVEs is small enough to enable the creation of a large database – see Remark 1.

Then, depending on the DoE and size of the database created, an appropriate machine learning algorithm needs to be selected. From the countless number of methods available in the literature, two methods are particularly prevalent [92]: Kriging and Neural Networks. Appendix C provides a broad overview of these methods but the interested reader is referred to the widely available literature on these subjects.

2. First example: finding general constitutive model of 2D hyperelastic composite by FEA

Depending on the problem to be solved, a large database may be constructed by high-fidelity computational simulations in a reasonable time frame without the need of using reduced order models to speed up each prediction. The overwhelming majority of the design examples reported in the literature [11,15,19,20,93,94] fall in this category where reduced order models are not needed since the simulations to obtain data are computationally inexpensive.

This first illustrative example is based on a problem posed by Yvonnet and co-workers [11–14] that focuses on finding the constitutive law of hyperelastic composites. This problem is generalized here with the goal of finding a constitutive law for two-dimensional hyperelastic composites reinforced by multiple elliptical particles, subjected to a significant variation of microstructural descriptors and large deformation. The following subsections demonstrate that the choice of the design of experiments method allied to the fact that the computational analysis of hyperelastic composites are relatively inexpensive to perform (a few seconds) allows the achievement of our goal without the need for a reduced order model.

2.1. Design of experiments (DoE)

As previously introduced, the DoE has to characterize the microstructure, material properties, and boundary conditions. For this problem the composite microstructures were generated according to the following criteria: (1) each material microstructure is periodic and composed of the same type of particles (same shape and size); (2) particles can assume an elliptical shape where the aspect ratio of the semi-axis ranges from 1 to 5; (3) particle volume fraction can change from 2% to 45%; (4) particles are allowed to overlap; (5) particle dispersion is controlled in such a way that the mean of nearest distances between particles' centers is within 0.3 mm and 0.5 mm; (6) particles have random orientation; and (7) particles and matrix are perfectly bonded. The undeformed length of the RVEs was considered to be 4 mm.

Considering the above criteria, four microstructure descriptors were selected with the respective bounds:

$$V_f = [2\%, 45\%], \quad N_p = [40, 100], \quad A_r = [1, 5], \quad \bar{r}_d = [0.3\ \text{mm}, 0.5\ \text{mm}] \tag{4}$$

where V_f is the particle volume fraction, N_p the number of particles, A_r the particles semi-axis aspect ratio, and $\bar{r}_d$ the mean of nearest distances measured at the particle centers. The above descriptors are subjected to the following constraints:

$$N_p < \frac{2L_c^2}{\sqrt{3\bar{r}_d^2}}, \quad a_{min} = 0.01\ \text{mm}, \quad b_{max} = 0.3\ \text{mm} \tag{5}$$

where the first constraint is the packing limit for an RVE with characteristic length L_c, while the other two constraints are for the size of the minimum minor semi-axis of the elliptical particles a_{min} and the maximum major semi-axis b_{max}.

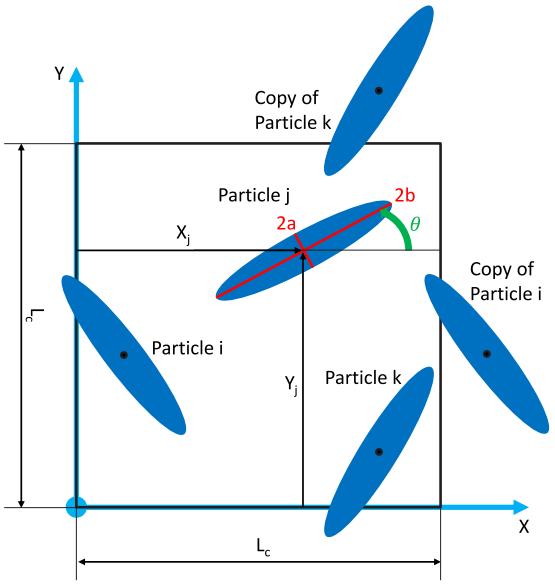


Fig. 2. Schematic of composite RVE geometry.

Fig. 2 presents a schematic of the microstructure. Each realization for each RVE is created by randomly generating the center position and semi-axis orientation for each elliptical particle and by excluding the particles that violate the above constraints.

For this problem only one specific material is considered for each constituent without changing its properties, i.e. there is no need to define material descriptors. The matrix material is a soft compressible elastomer modeled by the Arruda–Boyce [95] hyperelastic constitutive model, while the particles are a stiff compressible Neo-Hookean elastomer.

The Arruda–Boyce [95] constitutive law for the matrix material has a strain energy density function that depends on three polymer properties: the initial bulk modulus K_0, the initial shear modulus μ_0, and the stretch at which the polymer chain network becomes locked λ_m. The energy density function is then given by:

$$\psi\left(\mathbf{C}\right) = \mu \sum_{i=1}^{5} \alpha_i \beta^{i-1} \left(\hat{I}_1^i - 3^i\right) + \frac{K}{2}\left(\frac{J^2-1}{2} - \ln J\right) \tag{6}$$

where parameters $\beta = \frac{1}{\lambda_m^2}$, $\alpha_1 = \frac{1}{2}$, $\alpha_1 = \frac{1}{20}$, $\alpha_3 = \frac{11}{1050}$, $\alpha_4 = \frac{19}{7000}$, $\alpha_5 = \frac{519}{673750}$ are obtained using the first five terms of the inverse Langevin function, $J = det(\mathbf{F})$ is the Jacobian determinant of the deformation gradient $\mathbf{F}$, $\hat{I}_1 = I_1 J^{-2/3}$ depends on the first invariant $I_1 = Tr(\mathbf{C})$ of the right Cauchy–Green deformation tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, and parameter μ is calculated from the material properties λ_m and μ_0:

$$\mu = \frac{\mu_0}{2}\left(\sum_{i=1}^{5} i\alpha_i\beta^{i-1}\hat{I}_1^{i-1}\right)^{-1} = \mu_0\left(1 + \frac{3}{5\lambda_m^2} + \frac{99}{175\lambda_m^4} + \frac{513}{875\lambda_m^6} + \frac{42039}{67375\lambda_m^8}\right)^{-1}. \tag{7}$$

Note that the right Cauchy–Green deformation tensor $\mathbf{C}$ can be written as a function of the Green strain $\mathbf{E}$ as follows:

$$\mathbf{C} = 2\mathbf{E} + \mathbb{1} \tag{8}$$

with $\mathbb{1}$ being the identity matrix. For a review on the essentials of continuum mechanics and hyperelastic models see the book by Belytschko et al. [96].

The matrix material properties considered for the Arruda–Boyce model are:

$$K_0^{mat} = 800 \text{ MPa}, \quad \mu_0^{mat} = 180.5 \text{ MPa}, \quad \lambda_m^{mat} = 2.8. \tag{9}$$

The Neo-Hookean constitutive law for the particles is defined by the following strain energy density:

$$\psi\left(\mathbf{C}\right) = \frac{\mu_0}{2}\left(\hat{I}_1 - 3\right) + \frac{K_0}{2}\left(J-1\right) \tag{10}$$

where as before μ_0 is the initial shear modulus, K_0 is the initial bulk modulus, J is the Jacobian determinant, and $\hat{I}_1 = I_1 J^{-2/3}$.

The particle material properties considered for the Neo-Hookean model are:

$$K_0^{ptc} = 4.0 \text{ GPa}, \quad \mu_0^{ptc} = 1.9 \text{ GPa}. \tag{11}$$

For any hyperelastic model the second Piola–Kirchhoff stress $\mathbf{S}$ is directly related to the strain energy density:

$$\mathbf{S} = 2\frac{\partial \psi(\mathbf{C})}{\partial \mathbf{C}} = \frac{\partial \psi(\mathbf{E})}{\partial \mathbf{E}} \tag{12}$$

from which the expressions for the second Piola–Kirchhoff stress of each hyperelastic model can be trivially determined, see [96]. Similarly, the tangent modulus also called the second elasticity tensor $\mathbf{C}^{SE}$ is determined by computing the Hessian:

$$\mathbf{C}^{SE} = 4\frac{\partial^2 \psi(\mathbf{C})}{\partial \mathbf{C}\partial \mathbf{C}} = \frac{\partial^2 \psi(\mathbf{E})}{\partial \mathbf{E}\partial \mathbf{E}}. \tag{13}$$

The final descriptors needed for the DoE are given by the boundary conditions applied to the RVEs. Since the goal is to find the macroscopic constitutive law of the hyperelastic composite by homogenizing the micro-scale RVEs, the macroscopic strain measure of the material needs to be converted to a boundary value problem of the RVE.

Computational homogenization of multi-scale analysis has received significant attention in the last decade [60,97–100]. Broadly, computational homogenization can be classified as first-order and higher-order. First-order computational homogenization [97,99] assumes that the consecutive scales are separate; therefore, the macroscopic kinetic and kinematic quantities are obtained from a volume average of the micro-scale quantities of the RVE. Higher-order computational homogenization [98,100] does not assume scale separation; hence, the local kinematic and kinetic quantities of the RVE are homogenized within a finite region of the macroscopic scale by considering their Taylor series expansion with the desired number of higher-order terms, i.e. conserving higher gradients of the local fields of interest.

A first-order computational homogenization scheme is used herein, i.e. the characteristic length scale at which the macroscopic constitutive law is analyzed is assumed to be significantly larger than the domain of the RVE Ω. In this scheme the macroscopic deformation tensor $\bar{\mathbf{F}}$ is used to formulate a boundary value problem of the RVE, preferably periodic as discussed in [98],

$$\mathbf{u}\left(\mathbf{X}\right) = \left(\bar{\mathbf{F}} - \mathbb{1}\right)\cdot \mathbf{X} + \tilde{\mathbf{u}}\left(\mathbf{X}\right) \quad \text{on} \quad \partial\Omega \tag{14}$$

where $\mathbf{X}$ is the reference position vector on the boundary of the RVE $\partial\Omega$, $\mathbf{u}\left(\mathbf{X}\right)$ is the corresponding displacement and $\tilde{\mathbf{u}}\left(\mathbf{X}\right)$ signals the periodicity of the displacement field. If $\tilde{\mathbf{u}}\left(\mathbf{X}\right) = \mathbf{0}$ then uniform boundary conditions could be used instead, although these are not used here because multiple authors have concluded that the overall properties of the RVEs are better estimated using periodic boundary conditions [60,98,101–103].

In this problem the macroscopic constitutive law to be found relates the macroscopic Green strain $\bar{\mathbf{E}}$ (or equivalently, the macroscopic right Cauchy–Green deformation $\bar{\mathbf{C}}$) to the macroscopic second Piola–Kirchhoff stress $\bar{\mathbf{S}}$ via the macroscopic strain energy density $\bar{\psi}\left(\bar{\mathbf{E}}\right)$,

$$\bar{\mathbf{S}} = \frac{\partial \bar{\psi}(\bar{\mathbf{E}})}{\partial \bar{\mathbf{E}}} \tag{15}$$

and with the macroscopic tangent modulus given by

$$\bar{\mathbf{C}}^{SE} = \frac{\partial^2 \bar{\psi}(\bar{\mathbf{E}})}{\partial \bar{\mathbf{E}}\partial \bar{\mathbf{E}}}. \tag{16}$$

Therefore, the macroscopic deformation gradient $\bar{\mathbf{F}}$ necessary to impose the periodic boundary conditions from Eq. (14) needs to be obtained from the macroscopic Green strain $\bar{\mathbf{E}}$. As discussed by Yvonnet et al. [13], the macroscopic strain energy density $\bar{\boldsymbol{\psi}}$ is invariant under rotations $\bar{\mathbf{R}}$, so the macroscopic deformation gradient can be

computed by restricting the RVE rigid body rotation (i.e. $\bar{\mathbf{R}} = \mathbb{1}$) and then calculating the principal square root of the right Cauchy–Green deformation tensor,

$$\bar{\mathbf{F}} = \bar{\mathbf{U}} = \bar{\mathbf{C}}^{1/2} = \left(2\bar{\mathbf{E}} + \mathbb{1}\right)^{1/2} \tag{17}$$

from which the periodic boundary conditions of the RVEs can be defined by Eq. (14).

The solution of the boundary value problem is then obtained from the computational analyses of the different RVEs, from which the macroscopic quantities of interest are computed by homogenization of the microscopic quantities of interest. In particular to this problem, the quantity of interest is the strain energy density $\bar{\psi}$ as a function of the Green strain $\bar{\mathbf{E}}$. With this quantity it is possible to predict the second Piola–Kirchhoff stress $\bar{\mathbf{S}}$ of the RVE and compare it to the homogenization of the microscopic stress $\mathbf{S}$ over the RVE domain Ω.

As thoroughly explained by Yvonnet et al. [13], the homogenized stress measures have to be obtained from the first Piola–Kirchhoff stress $\mathbf{P}$ due to the Hill–Mandel lemma,

$$\langle \mathbf{P} : \mathbf{F} \rangle = \langle \mathbf{P} \rangle : \langle \mathbf{F} \rangle = \bar{\mathbf{P}} : \bar{\mathbf{F}} \tag{18}$$

where $\langle \bullet \rangle$ indicates a volume average integral over the undeformed RVE domain Ω_0. Hence, the homogenization of the second Piola–Kirchhoff stress $\mathbf{S}$ is computed from the first Piola–Kirchhoff stress as follows,

$$\bar{\mathbf{S}} = \bar{\mathbf{F}}^{-1} \cdot \bar{\mathbf{P}}. \tag{19}$$

To complete the DoE step, the range of macroscopic Green strains $\bar{\mathbf{E}}$ needs to be defined. These strains are then converted to the macroscopic deformation gradient from Eq. (17) so that the periodic boundary conditions can be applied to the RVEs via Eq. (14). The solution to the boundary value problem of the RVEs is then found from the computational analyses, from which the subsequent homogenization of the strain energy density is performed.

The bounds of the three macroscopic Green strain components considered herein are:

$$\bar{E}_{11} = \bar{E}_{22} = [-0.1, 1.5], \quad \bar{E}_{12} = [-0.3, 0.3] \tag{20}$$

under the following constraint:

$$-1 \leq \frac{2\bar{E}_{12}}{\sqrt{(2\bar{E}_{11}+1)(2\bar{E}_{22}+1)}} \leq 1 \tag{21}$$

due to the limitation on the angle between the stretch directions of the Green strain arising from finite deformation theory.

Fig. 3 shows a DoE obtained from Sobol sequence [32,33] for the three strain components with the bounds given in (20) and under the constraint given in (21) without considering additional descriptors, i.e. with the following input variables $x_1 \equiv \bar{E}_{11}$, $x_2 \equiv \bar{E}_{22}$, and $x_3 \equiv \bar{E}_{12}$. A DoE with more descriptors, for example the 4 microstructural descriptors previously listed plus these 3, is obtained in a similar way but the input variables domain has dimension 7. This figure illustrates three useful characteristics of Sobol sequence:

1. The sample points are spread out non-uniformly over the input variables space;
2. There are no coincident projections of the sample points in the different hyperplanes of the input variables space;
3. The input variables space is successively refined as the Sobol sequence progresses.

The first point is connected with the second, since the non-uniformity of this space filling design is associated with the fact that all the points are distinguishable in any of the projection planes of the descriptors – see Fig. 3. This occurs for a DoE of any dimension generated by Sobol sequence. Note that if a regular grid of sample points was used instead, there would be multiple coincident point projections in the different hyperplanes which deteriorates the machine learning and metamodeling process [27,29,92].

The third point is significantly useful in practice and is illustrated by the different colors for the 1000 DoE points shown in Fig. 3. The first 50 points in this Sobol sequence are shown in red, where it is clear that the points are spread out through the input variables space. The next 150 points are shown in green, demonstrating that the space is successively refined by points that never coincide with the previous points. The remaining 800 points (in blue) also clearly show this successive refinement, always guaranteeing a space filling design. This is useful because one

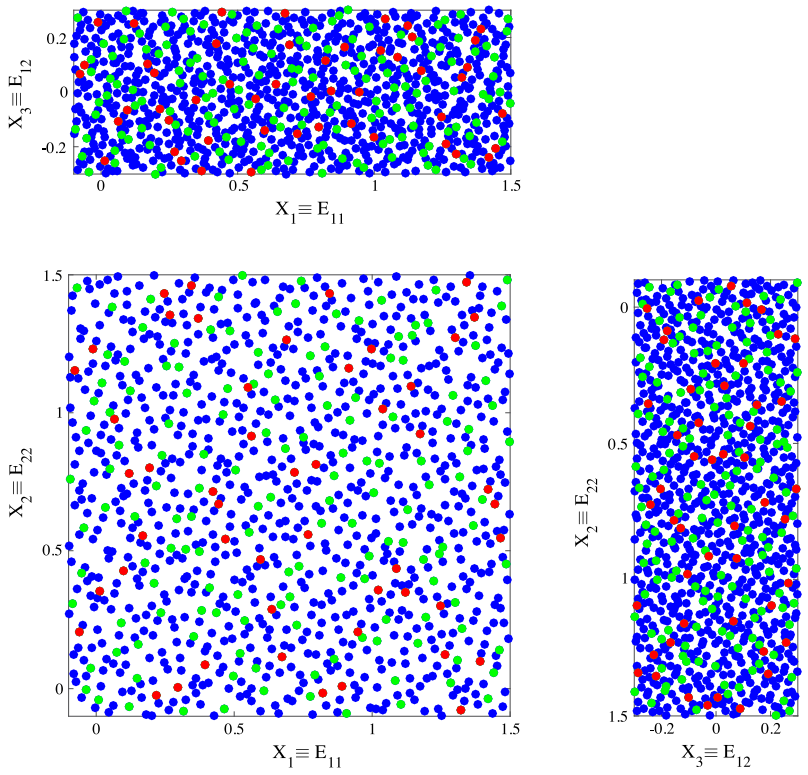


Fig. 3. Design of experiments with 1000 points for the three boundary condition descriptors using Sobol sequence. Red dots indicate the first 50 points, green dots indicate the next 150 points, and blue dots the remaining 800 points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

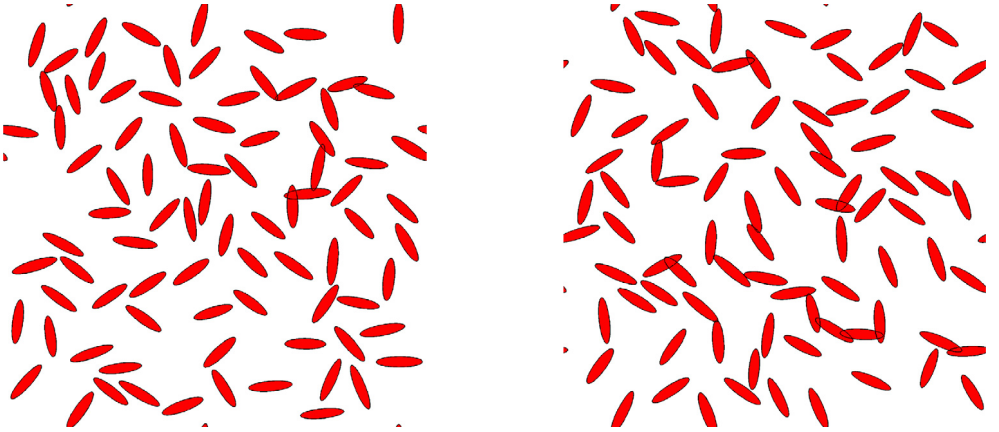
does not know *a priori* how many sample points in the DoE lead to a good model determined by the machine learning procedure. Therefore, consecutive subsets of a DoE will be space-filling designs themselves. The consecutive increase of the number of DoE points in the sequence is then associated to a global refinement of the design space, exploring regions that were previously unexplored.

2.2. *Computational analysis and machine learning*

The created DoE uses a space filling design and no prior knowledge of the hyperelastic composite RVEs is assumed. In reality, for this illustrative problem it would be possible to simplify the machine learning process by reducing the dimension of the DoE. For example, the constitutive models of the two phases of the composite only depend on the two invariants of the Green strain, i.e. only the two principal components of the Green strain actually influence the constitutive behavior. This means that the shear component is redundant since one can always rotate to the frame of principal strains.

Most microstructural descriptors considered are also expected to have a small influence on the response, except the volume fraction. Volume fraction affects the response significantly because the reinforcements are stiffer than the matrix and the average response of the composite is mostly related to the amount of reinforcement embedded in the matrix, not to its particular shape. The other microstructural descriptors influence the strain concentration around the particles but since there is no plasticity or damage in this problem those effects are largely eliminated after homogenization.

Nevertheless, this example provides an important basis to assess the capabilities of the data-driven framework applied to materials design. The models found from machine learning should be able to characterize the response of the material accurately, but also to inform about the influence of the different descriptors on the response. The following three step approach is proposed when first applying the data-driven framework to a specific problem.



(a) Realization 1 (b) Realization 2

Fig. 4. Two different realizations of the microstructure for the following fixed set of microstructural descriptors: $V_f \approx 17\%$, $N_p = 84$, $A_r \approx 4.1$, and $\bar{r}_d \approx 0.32$ mm. Note that the matrix material is not shown for clarity, and that both microstructures are periodic.

2.2.1. *Step 1: uncertainty quantification*

As previously described, after the completion of the DoE the representative microstructures are generated for each DoE point so that the computational analyses can be performed, recall Fig. 1. However, a fixed set of microstructural descriptors may not uniquely characterize a material microstructure. In fact, for the 7 dimension DoE used in this example – see (4) and (20) – one can fix all the microstructural descriptors (volume fraction, number of particles, elliptical semi-axis aspect ratio, and mean of nearest distances) and still generate multiple realizations that have different particle positions and orientations – see Fig. 4.

The first step of the procedure is then to quantify the uncertainty of the response of each microstructure for a range of other descriptors (boundary conditions and/or material properties). The length scale of the microstructure domain L_c is strongly associated with the uncertainty of the response. A larger L_c tends to eliminate the stochastic effects which implies that the domain is representative (RVE), while a smaller L_c leads to structures that have different responses (SVEs). Naturally, larger domains imply greater computational cost per simulation — see Remark 2.

Remark 2. The analyst needs to decide whether to conduct less expensive analyses for multiple realizations of each DoE point (multiple SVEs) and subsequently average them, or to simulate one realization per DoE point (RVE) knowing that those simulations have higher computational cost.

In this example the characteristic length L_c of the microstructure is considered large enough if the average coefficient of variation of the homogenized strain energy density $\bar{\psi}\left(\bar{\mathbf{E}}\right)$ and of the Euclidean norm of the homogenized second Piola–Kirchhoff stress $\|\bar{\mathbf{S}}\| = \sqrt{\bar{S}_{11}^2 + \bar{S}_{22}^2 + \bar{S}_{12}^2}$ are approximately less than 5%.

In general, the maximum coefficient of variation criterion for any average quantity of interest $\bar{q}_i$ of the RVE is then written as,

$$max\left[CV\left(\bar{q}_i^{(m,p,l)}\right)\right] \lesssim 5\%, \quad \forall_{m,p,l} \tag{22}$$

where $\bar{q}_i^{(m,p,l)}$ is the homogenized quantity of interest for microstructure m, property set p and boundary condition l. The coefficient of variation is defined as,

$$CV\left(\bar{q}_i^{(m,p,l)}\right) = \frac{\sigma\left(\bar{q}_i^{(m,p,l)}\right)}{\bar{\bar{q}}_i^{(m,p,l)}} \tag{23}$$

Table 1
Four microstructures/RVEs for uncertainty quantification.

	V_f	N_p	A_r	$\bar{r}_d$
RVE 1	4.7%	47	4.9	0.50 mm
RVE 2	17.1%	84	4.1	0.32 mm
RVE 3	29.5%	48	1.3	0.49 mm
RVE 4	39.0%	84	1.4	0.33 mm

where $\sigma\left(\bar{q}_i^{(m,p,l)}\right)$ is the standard deviation of all the realizations R for each microstructure m, property set p and boundary condition l,

$$\sigma\left(\bar{q}_i^{(m,p,l)}\right) = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[\left(\bar{q}_i^{(r,m,p,l)} - \bar{\bar{q}}_i^{(m,p,l)}\right)^2\right]} \tag{24}$$

and $\bar{\bar{q}}_i^{(m,p,l)}$ is the mean of the homogenized quantity of interest for all the realizations R,

$$\bar{\bar{q}}_i^{(m,p,l)} = \sum_{r=1}^{R}\frac{\bar{q}_i^{(r,m,p,l)}}{R} \tag{25}$$

with $\bar{q}_i^{(r,m,p,l)}$ being the homogenized quantity of interest for realization r of the microstructure m, property set p and boundary condition l.

The coefficient of variation is meaningful for non-negative quantities of interest, as for the strain energy density and the norm of the components of the second Piola–Kirchhoff stress, but not for the individual stress components. So, in this case $\bar{q}_1 \equiv \bar{\psi}\left(\bar{\mathbf{E}}\right)$ and $\bar{q}_2 \equiv \|\bar{\mathbf{S}}\|$.

The estimation of the uncertainty of the response of each microstructure was then performed by generating $R = 20$ realizations for each of the $M = 4$ different microstructures described in Table 1. All the microstructures were subjected to the same $L = 50$ deformation states obtained by Sobol sequence, similar to the randomization shown in Fig. 3. As mentioned in the previous section, only one set $P = 1$ of material properties was considered for the particles and matrix in this problem, see Eq. (9) for the matrix properties and Eq. (11) for the particles.

Remark 3. The estimation of uncertainty of this problem involved a total of $R \times M \times P \times L = 20 \times 4 \times 1 \times 50 = 4000$ simulations for each characteristic length of the microstructure L_c. The largest characteristic length simulated was $L_c = 4$ mm, corresponding to an approximate computation time of 0.5 min for generating each finite element mesh and 4 min to perform the respective analysis. The output files of the 4000 analyses occupied more than 80 GB of storage space in the high performance computing cluster, and the simulations were conducted in parallel with only 1 processor each. These output files were subsequently homogenized to produce the database for the uncertainty quantification analysis.

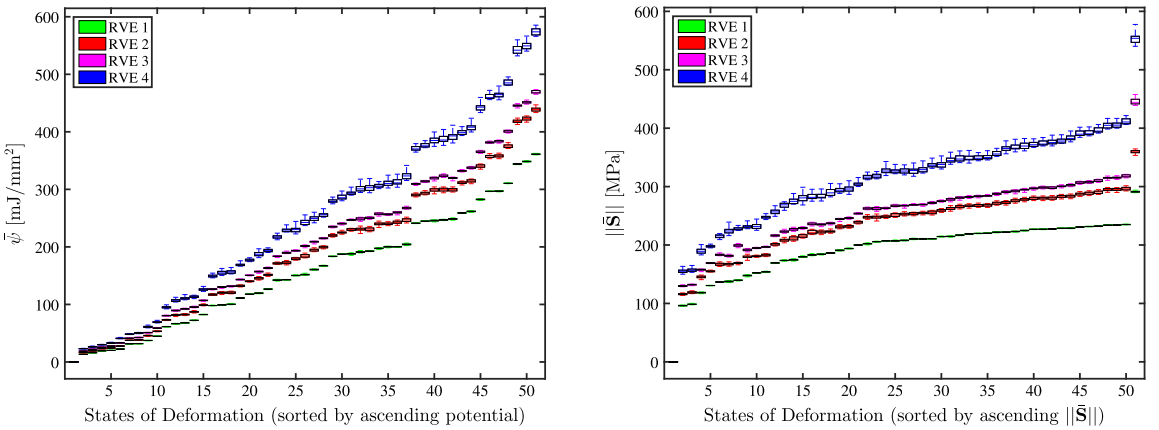
Figs. 5(a) and (b) illustrate the uncertainty for each macroscopic deformation state, i.e. each boundary condition l, applied to the different realizations of the 4 microstructures for the largest characteristic length $L_c = 4$ mm that approximately satisfied criteria (22). For clarity, the states of deformation in both figures were sorted such that the quantities of interest are in ascending order. The figures include a box for each deformation state and microstructure where the central mark is the median of the quantity of interest, the edges of the box are the 25th and 75th percentiles, and the top and bottom dashed lines represent the maximum and minimum value of the quantity of interest, respectively.

The maximum coefficient of variation of the macroscopic strain energy density $\bar{\psi}$ and of the norm of the second Piola–Kirchhoff stress $\|\bar{\mathbf{S}}\|$ for each microstructure with $L_c = 4$ mm is shown in Table 2.

The uncertainty of the response for each microstructure shows that the stress measure has higher uncertainty than the strain energy density, as expected. This follows directly from the fact that the stresses are derivatives of the strain energy density with respect to the strain measure, as pointed out previously, increasing the uncertainty. Moreover, observing Figs. 5(a) and (b) it is clear that the microstructures with higher volume fraction lead to a higher uncertainty, which is also expected [43,104,105].

Table 2
Uncertainty quantification of strain energy density and PK2 stress of four RVEs.

	$max\left[CV\left(\bar{\psi}^{(m,p,l)}\right)\right]$	$max\left[CV\left(\|\bar{\mathbf{S}}^{(m,p,l)}\|\right)\right]$
RVE 1	0.6%	1.3%
RVE 2	1.4%	2.9%
RVE 3	1.0%	2.4%
RVE 4	2.3%	5.5%



(a) Uncertainty of strain energy density.

(b) Uncertainty of the norm of second Piola–Kirchhoff stress.

Fig. 5. Uncertainty quantification of (a) strain energy density and (b) norm of the stress components for 4 RVEs with $L = 4$ mm subjected to the first 50 deformation states obtained from the Sobol sequence. Note that the deformation states were sorted such that the potential energy (a) and the norm of the stress components in (b) are in ascending order. The descriptors of the 4 RVEs are included in Table 1.

This initial estimation of uncertainty for the problem before evaluating a larger input variables space with all the descriptors is important when the problem under analysis involves large quantities of data. Since the uncertainty is sufficiently low, the machine learning process can continue using a single realization for each microstructure, which decreases significantly the number of simulations necessary to determine the material constitutive law for a large number of boundary conditions and microstructures.

2.2.2. Stage 2: machine learning for a single RVE

Having found that a single realization for each microstructure with characteristic length $L_c = 4.0$ mm characterizes reasonably well the response for any boundary condition, it is now useful to estimate the convergence obtained in the approximation of the composite constitutive behavior. Before solving the problem for the DoE of largest dimension with varying microstructure and boundary conditions, it is useful to analyze in detail the response for a typical microstructure. This preliminary analysis provides information on the adequacy of the space-filling design for the boundary conditions, showing if local refinements are necessary and giving an estimate for the number of DoE points needed to predict the composite response within a certain accuracy.

Hence, a single RVE for a specific microstructure is considered under a larger number of boundary conditions than the ones used for uncertainty quantification. The selected RVE has the following microstructural descriptors:

- RVE 5: $V_f \approx 21.0\%$, $N_p = 60$, $A_r \approx 3.8$, and $\bar{r}_d \approx 0.4$ mm.

A DoE with 1000 points was created from a Sobol sequence where each DoE point corresponds to a different boundary condition applied to the RVE, while the remaining descriptors are fixed. One thousand simulations of RVE 5 are conducted and the homogenized strain energy density is approximated by the two machine learning methods introduced previously: kriging and neural networks. The three stress components are then determined by differentiating the approximated homogenized strain energy density with respect to the Green strain components.

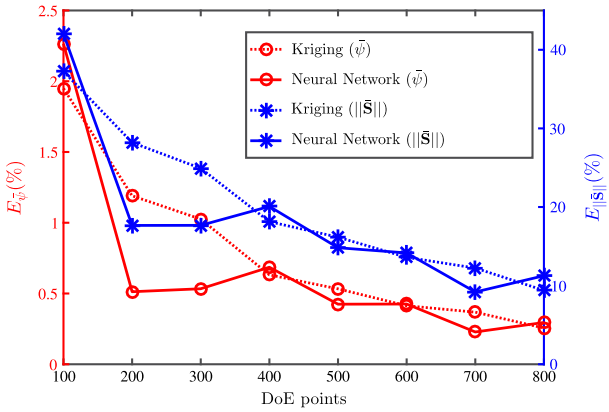


Fig. 6. Error of the strain energy density and the norm of the second Piola–Kirchhoff stress predicted by kriging and neural networks models of RVE 5 when considering different DoE sizes.

As for Kriging, the so-called Gaussian correlation function was used, see Eq. (C.3), to render the fitted model infinitely differentiable since at least the first and second derivates are needed. All the model parameters of the Kriging model were estimated by maximizing the multivariate Gaussian likelihood function, see Eq. (C.5), using a multi-start gradient-based optimization technique. For the neural networks a variety of network architectures were tested, e.g. cyclic or acyclic with single or multiple hidden layers, and it was found that a feedforward network with a single hidden layer provides the most parsimonious yet accurate structure for the two illustrative problems presented herein. The optimum number of neurons in the single hidden layer and the model parameters (weights and bias terms) were estimated via, respectively, 20-fold cross-validation (CV) and the backpropagation algorithm.

In addition, an error metric needs to be defined to estimate the accuracy of the approximation for the quantities of interest determined from each machine learning method. Relative error metrics should be defined for non-negative quantities, so the error of the stress predictions is calculated for N points not included in the training data as follows,

$$E_{\bar{\mathbf{S}}} = \frac{1}{N} \sum_{(m,p,l)=1}^{N} \frac{\|\hat{\bar{\mathbf{S}}}^{(m,p,l)} - \bar{\mathbf{S}}^{(m,p,l)}\|}{\|\bar{\mathbf{S}}^{(m,p,l)}\|} \tag{26}$$

where N is the number of data points used for validation, $\hat{\bar{\mathbf{S}}}^{(m,p,l)}$ is the predicted homogenized stress for microstructure m, property set p and boundary condition l, and $\bar{\mathbf{S}}^{(m,p,l)}$ is the observed value from the actual finite element analysis of the RVE. Note that each point is labeled as (m, p, l), corresponding to a particular microstructure, property set and boundary condition.

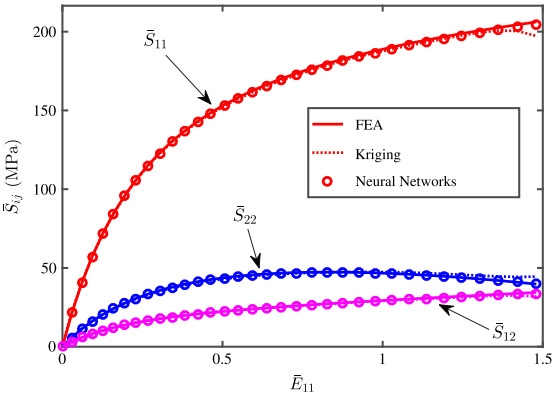
The error of the strain energy density is defined similarly,

$$E_{\bar{\psi}} = \frac{1}{N} \sum_{(m,p,l)=1}^{N} \frac{\left|\hat{\bar{\psi}}^{(m,p,l)} - \bar{\psi}^{(m,p,l)}\right|}{\left|\bar{\psi}^{(m,p,l)}\right|}. \tag{27}$$

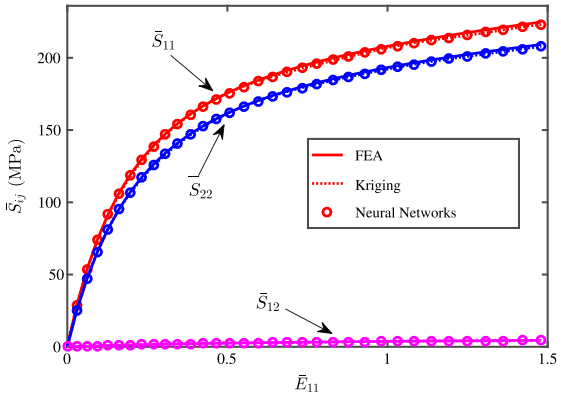
In this case the models are determined for the same microstructure and property set, so m and p do not change and all the DoE points correspond to changing the boundary conditions l.

Convergence for this problem is estimated by determining different kriging and neural network models for an increasing number of successive DoE points in increments of one hundred until a total of eight hundred points. The last two hundred points of the initial DoE are used for validation, i.e. $N = 200$ in Eqs. (26) and (27).

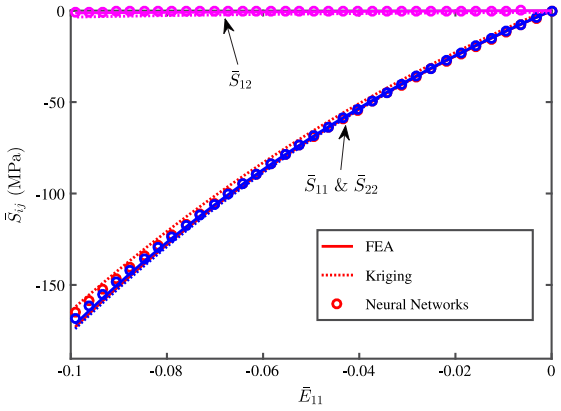
Fig. 6 shows the errors of $\bar{\psi}$ and $\|\bar{\mathbf{S}}\|$ obtained for both methods using different DoE sizes. From the figure it is clear that both methods lead to similar approximation errors for the strain energy density (below 0.5%) and stresses (approximately 10%). The strain energy density is accurately approximated even considering a small DoE. However, the second Piola–Kirchhoff stress of the RVE for different deformation states requires a larger DoE since it is related to the first derivative of the strain energy density. As expected, considering more DoE points in the machine learning process leads to more accurate predictions.



(a) Final deformation state: $\bar{E}_{11} = 1.5$, $\bar{E}_{22} = -0.1$ and $\bar{E}_{12} = 0.3$.



(b) Final deformation state: $\bar{E}_{11} = 1.5$, $\bar{E}_{22} = 1.0$ and $\bar{E}_{12} = 0$.



(c) Final deformation state: $\bar{E}_{11} = -0.1$, $\bar{E}_{22} = -0.1$ and $\bar{E}_{12} = 0$.

Fig. 7. Comparison of the stress–strain response of RVE 5 from the undeformed configuration until three different final deformation states. The kriging model shown in a dashed line is obtained for 600 DoE points, while the neural networks model (circles) is obtained for 800 DoE points.

The largest errors shown in Fig. 6 occur for highly localized regions of the deformation space, in particular for large compression combined with large shear. These correspond to regions where the two principal Green strains are significantly negative (below -0.1) and where the deformation space is not sampled adequately. Recall Fig. 3 where it is clear that even for a DoE of one thousand points the compressive region is sampled only with a few points due to the small size of this region. Local refinement of the DoE in this region reduces the error of the models, as shown in the next subsection for the largest DoE analyzed herein.

Figs. 7(a), (b) and (c) show the approximations obtained when deforming the RVE for three loading paths that evolve linearly from the undeformed configuration until different limits of the deformation space, see (20). For clarity all stress components are plotted as a function of the largest deformation strain E_{11}. Since the approximations obtained by kriging and neural networks are very similar when considering the same number of DoE points, the results for the three deformation paths are shown for the kriging model considering six hundred DoE points and for the neural networks model considering eight hundred points. Otherwise, the results from the two methods are difficult to distinguish in the entire deformation space.

The agreement between both models and the finite element analyses of the RVE is good for all three cases. Note that the models are determined for the scattered points of the DoE and that these do not coincide with the predicted points in the figures. Particular attention should be given to Fig. 7(a) that shows the result obtained for the largest tensile deformation along direction 1, the largest compression along direction 2 and greatest shear. It can be seen that

Table 3
Sensitivity analysis for RVE 5 (rounded to third decimal place).

	$\bar{E}_{11}$	$\bar{E}_{22}$	$\bar{E}_{12}$
S_i	0.504	0.483	0.002
S_{T_i}	0.514	0.493	0.007

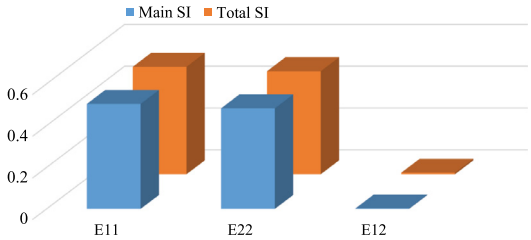


Fig. 8. Sensitivity indices of the kriging model of RVE 5 obtained for 1000 DoE points. The values of the sensitivity indices are shown in Table 3.

for points closer to the bounds of the deformation space the solution obtained with six hundred DoE points is less accurate.

In addition, important information about the influence of the inputs on the model outputs can be obtained via global sensitivity analysis, e.g. variance based methods [106,107]. In variance-based sensitivity analysis there are two indices that are typically used to measure sensitivity: the first-order or main sensitivity index S_i; and the total effect or total sensitivity index S_{T_i}. Appendix D offers a brief description on how these indices can be determined. Both indices are within [0, 1] and indicate the influence of the corresponding input on the output (no influence if 0; strong influence if 1). For any input variable x_i, the inequality $S_i \leq S_{T_i}$ is satisfied, and the difference between the two measures quantifies how much interaction x_i has with other inputs.

Fig. 8 and Table 3 show the sensitivity indices calculated for the kriging model using the entire DoE of one thousand points. As can be observed the sensitivity of the response is basically the same for E_{11} and E_{22} since the RVE is isotropic and the bounds are symmetric. Interestingly, the sensitivity to E_{12} is negligible which shows that for the majority of the deformation space the influence on the response is small. This may be related to the fact that E_{12} is not necessary to determine the constitutive law, since there are only two invariants of deformation.

2.2.3. Step 3: machine learning for the general problem

Finally, after estimating the uncertainty associated with different microstructures and the number of boundary conditions that are necessary to reasonably determine the constitutive law of a single microstructure, the complete analysis for the 7 descriptors previously discussed can be conducted, see Eqs. (4) and (20).

Taking advantage of the fact that the Sobol sequence does not have coincident DoE points projected in the different hyperplanes, a possible way to minimize the amount of DoE points is to create a seven dimension DoE where each point corresponds to a different combination of all the descriptors. In practice this means the creation of one finite element mesh for each distinct RVE of the microstructure and each distinct boundary condition applied to that RVE. The alternative would be to do a cross-design, where each microstructure would be subjected to the same boundary conditions as was done previously for the uncertainty quantification analyses. A cross-design is particularly inefficient [92], especially with the kriging method due to the need to invert the covariance matrix which can cause singularity issues.

Testing hundreds of thousands or even millions of DoE points is possible, but that may be an unnecessary effort since the sensitivity of the response to the different descriptors is unknown and there may be redundancy. A DoE of 10,000 points is used such that the database could be created in less than one day of computation (an average of fifty processors were used simultaneously; the database from the RVE analyses had an approximate size of 0.5 TB). This number of DoE points is sufficiently large to limit the use of kriging because the inverse of the covariance matrix consumes too much computer memory using our non-optimized MATLAB® code. Hence, machine learning is performed herein only using neural networks.

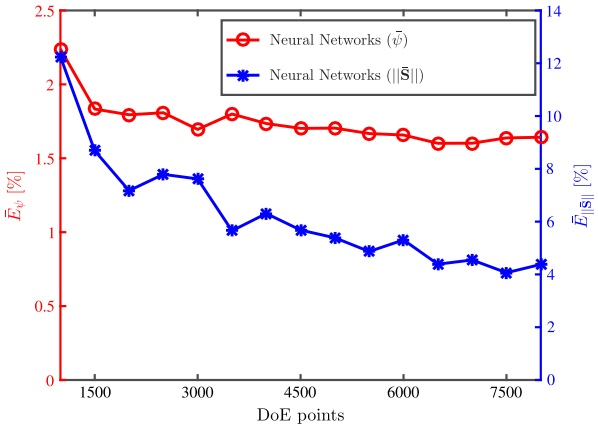


Fig. 9. Error of the strain energy density and the norm of the second Piola–Kirchhoff stress predicted by neural networks models of the 7 dimension DoE. Note that, as before, stresses are directly calculated from differentiating the strain energy density approximated from neural networks.

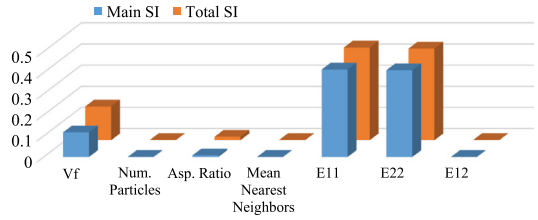


Fig. 10. Sensitivity indices of the neural networks model of the 7 dimension DoE with 10,377 points. The values of the total sensitivity indices are shown in Table 4.

In addition to the 10,000 DoE points that define the space-filling design, a local refinement of the input variables space is included in the highly localized regions where the prediction errors occurred. This local refinement was obtained such that the optimality of the Sobol sequence would not be lost. The procedure was as follows:

1. 20,000 points were generated from a Sobol sequence of dimension 7, and the first 10,000 points were included in the DoE;
2. Of the remaining 10,000 points the only ones that were included in the DoE obeyed the following constraints:
   - $V_f > 30\%$
   - $V_f > 40\%$, or $\bar{E}_{12} < -0.2$ or $\bar{E}_{12} > 0.2$
   - $\bar{E}_{11} < 0$ or $\bar{E}_{22} < 0$ or $\bar{E}_{12} < -0.25$ or $\bar{E}_{12} > 0.25$.

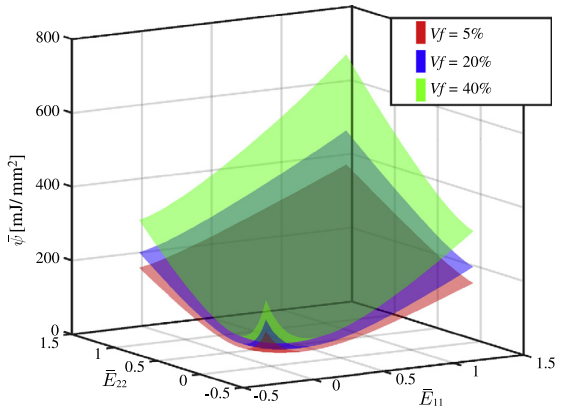
These constraints correspond to regions with large shear and volume fraction, as well as regions with compression and large shear. Of the last 10,000 of the 20,000 points only 377 satisfied the above constraints. Therefore, the entire DoE included 10,377 points which led to a significant decrease of the overall error at those localized regions, as seen in Fig. 9 when compared to Fig. 6.

Fig. 9 presents the convergence of the neural networks models obtained by successively increasing the size of the DoE. The last 2500 DoE points are used for validation of the models. A similar trend to the single RVE problem is observed, where the strain energy density is accurately predicted even for a relatively small number of DoE points, while the prediction from the second Piola–Kirchhoff stress requires a larger DoE. These results are acceptable considering the uncertainty associated to each RVE and that a large input variables space is being explored.

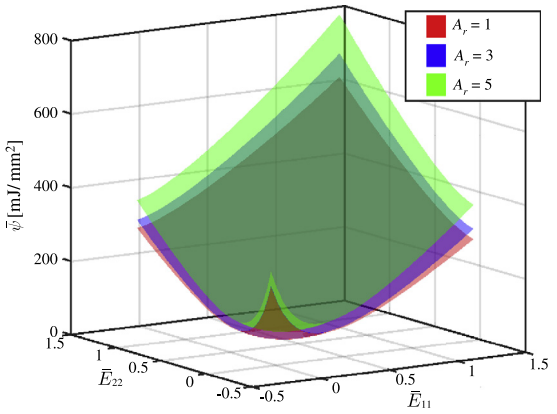
The sensitivity indices determined for the DoE with 10,377 points are shown in Fig. 10 and Table 4. The influence of particle volume fraction on the response is very significant, as expected. The number of particles and their dispersion, at least within the bounds considered in this problem, have small impact on the response. The aspect ratio of the particles has a noticeable influence, though inferior to the influence of the volume fraction. As observed

Table 4
Sensitivity analysis for 7 dimension DoE (rounded to the third decimal place).

	V_f	N_p	A_r	$\bar{r}d$	E_{11}	E_{22}	E_{12}
S_i	0.117	0.000	0.007	0.000	0.414	0.411	0.000
S_{T_i}	0.160	0.001	0.016	0.001	0.439	0.434	0.001



(a) Influence of particle volume fraction V_f on potential energy ($N_p = 60$, $A_r = 3$, $\bar{r}_d = 0.4$ mm, $E_{12} = -0.3$).



(b) Influence of particle aspect ratio A_r on potential energy ($V_f = 0.4$, $N_p = 60$, $\bar{r}_d = 0.4$ mm, $E_{12} = -0.3$).

Fig. 11. Variation of potential energy as a function of different (a) particle volume fractions and (b) particle aspect ratios, while maintaining the remaining descriptors fixed.

before, the shear component of the strain also has a negligible influence on the global response. These general trends are in part expected, as discussed in Section 2.2, but the actual contribution and the relative importance of the different descriptors would be challenging to predict *a priori*.

Fig. 11 illustrates the dependence of the macroscopic strain energy density on (a) the volume fraction and (b) the aspect ratio of the particles for a particular set of descriptors. Fig. 11(a) reinforces the findings that the volume fraction has the most significant influence on the macroscopic strain energy density for the entire range of deformation states, while Fig. 11(b) shows that the variation caused by the aspect ratio of the particles is less pronounced.

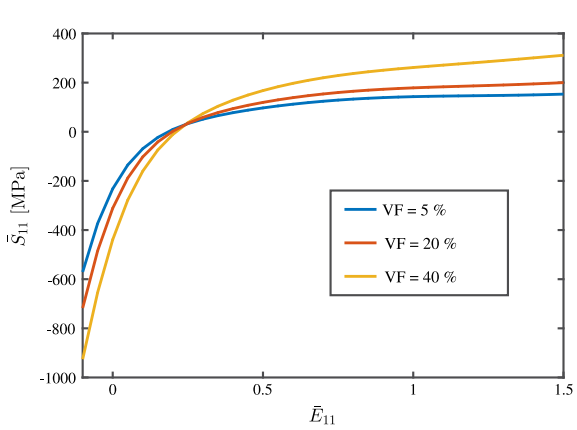
The predictions for the stress–strain behavior of the composite for components 11 under the same descriptors and for $\bar{E}_{22} = 0$ are shown in Fig. 12. Once again, the influence of volume fraction and aspect ratio is illustrated by these figures. The effect of the aspect ratio is in part given by the fact that some long elliptical particles are forced to bend when the RVE is deformed.

3. Second example: design of 3D inelastic composite via self-consistent clustering analysis

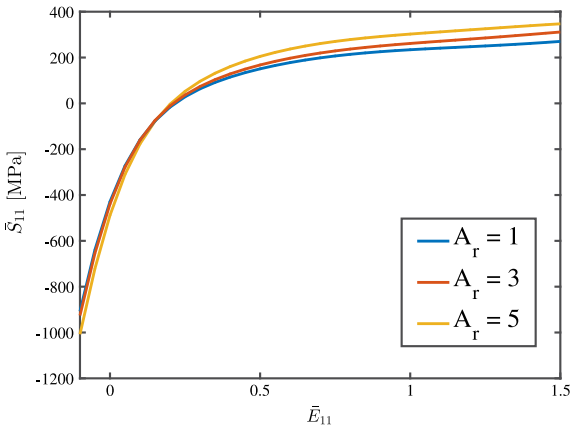
The previous example and other design examples found in the literature for linear and nonlinear elastic micro- and macro-structures [11,15,19,20,93,94] are greatly simplified by the fact that they are computationally inexpensive. However, when modeling irreversible processes such as plasticity and fracture the computational expense of each representative volume element rises dramatically. In order to address this issue, the authors recently proposed a new method called self-consistent clustering analysis (SCA) – see [46] and Appendix B – that can be used instead of the direct numerical simulations (DNS) of high-fidelity RVEs.

A three-dimensional RVE is considered for a particle reinforced composite with a particle volume fraction of 20%. The same microstructure was used in the RVE introduced in [46]. The RVE has 80 mm of length and each spherical particle has a radius of 13.5 mm. The high-fidelity RVE is analyzed by the finite element method using a mesh of 512,000 elements that was shown previously [46] to lead to similar results as a coarser mesh. The RVE analysis is considered as the reference solution, and here both phases of the composite are assumed inelastic following the von Mises (J_2) elasto-plastic model. Recall that the yield surface of this model is given by,

$$f = \sigma_{vm} - \sigma_Y(\varepsilon^p) \leqslant 0 \tag{28}$$



(a) Influence of particle volume fraction V_f on stress–strain behavior for component 11 ($N_p = 60$, $A_r = 3$, $\bar{r}_d = 0.4$ mm, $E_{22} = 0$, $E_{12} = -0.3$).



(b) Influence of particle aspect ratio A_r on stress–strain behavior for component 11 ($V_f = 0.4$, $N_p = 60$, $\bar{r}_d = 0.4$ mm, $E_{22} = 0$, $E_{12} = -0.3$).

Fig. 12. Variation of potential energy as a function of different (a) particle volume fractions and (b) particle aspect ratios, while maintaining the remaining descriptors fixed.

with $\sigma_{vm} = \sqrt{3J_2}$ being the von Mises equivalent stress dependent on the second invariant of the deviatoric stresses J_2, and σ_Y being the yield stress determined from the respective hardening law.

Hence, the matrix material (labeled as phase 1) has the following properties:

$$E_1 = 100 \text{ GPa}, \quad \nu_1 = 0.3 \tag{29}$$

with the hardening law given by:

$$\sigma_{Y_1}(\varepsilon_1^p) = 100 + 300\left(\varepsilon_1^p\right)^{0.4} \text{ MPa} \tag{30}$$

where ε_1^p is the equivalent plastic strain of the matrix phase.

The properties of the reinforcement material (labeled as phase 2) are:

$$E_2 = 500 \text{ GPa}, \quad \nu_2 = 0.19 \tag{31}$$

with the hardening law given by:

$$\sigma_{Y_2}(\varepsilon_2^p) = a + b\left(\varepsilon_2^p\right)^{0.2} \text{ MPa} \tag{32}$$

where ε_2^p is the equivalent plastic strain of the particle phase, while a and b are the two input design variables for the problem. Recall that parameter a is related to yielding, while b is related to hardening of the particles.

The goal of this illustrative design problem is to find the properties a and b of the particle phase such that the toughness of the composite is maximized. Toughness is defined as the integral of the stress–strain curve obtained from a uniaxial tension test of the composite material before fracture. A simplified fracture model is considered for the composite material:

- The composite RVE fails when 10% of the matrix phase has a maximum strain component above 0.07;
- No damage model is implemented, i.e. this composite is considered brittle;
- The particles are considered to fail at a significantly higher strain level, i.e. the composite fails uniquely by matrix failure.

The above considerations were made so that the results could be easily interpreted for this illustrative example. Any fracture model could have been implemented, and one or more damage laws could have been considered. However, this would increase the number of model parameters without any benefit for a first demonstration of the framework.

Maximizing material toughness is the result of two competing factors: ductility and strength. If the particle phase has high yield strength (remaining elastic) the overall strength of the composite increases but its ductility decreases

because the matrix phase is highly strained and fails prematurely. On the other hand, if the particles have low yield strength and hardening but high ductility the matrix material is less strained for the same overall composite strain, which increases ductility of the composite at the expense of decreasing its strength. Finding a compromise between strength and ductility of the reinforcement particles of the composite is expected to lead to a maximum composite toughness.

Note that since the problem involves the assessment of the local (matrix) strains to determine fracture in a three-dimensional composite with both phases being inelastic, the solution to this problem would be difficult to obtain without a data-driven framework such as the one proposed herein.

The approach to this problem is similar to the one outlined in the previous section. The only difference is that the predictions of each data point are not conducted directly from the high-fidelity RVE. Instead, the high-fidelity RVE is loaded in 6 orthogonal loading conditions within the elastic regime in order to complete the offline stage of SCA, so that SCA can then be used to predict the behavior of the reduced RVE under plasticity up to fracture. Elastic simulations of the high-fidelity RVE have negligible computation time, while the complete analysis considering plasticity would take 72 hours as discussed next.

Without assuming any prior knowledge about this problem, it can be useful to sample a large part of the design space with fast predictions (small number of SCA clusters). If the accuracy of these fast predictions is reasonable, the global trend of the response (toughness) can be quickly captured as a function of the input variables (a and b). Evidently, the SCA predictions should be validated by comparing to the high-fidelity predictions for different input variables to assess the accuracy and convergence of the method. As shown next, multiple reduced RVEs are considered where different number of SCA material clusters were chosen. The number of clusters of every reduced RVE in the particle phase (phase 2) is related to the number of clusters in the matrix phase (phase 1) by,

$$k_2 = \lceil k_1/4 \rceil \tag{33}$$

where k_1 is the number of matrix clusters and k_2 is the number of particle clusters in the reduced RVE.

Considering 64 matrix clusters (hence, 16 particle clusters) it takes 10 minutes to complete the offline stage of SCA and an average of 27 seconds to perform the online stage for each design point.[6] The total computation time of the SCA method is obtained by adding the one-time offline computation time to the computation time of all the online analyses conducted (including postprocessing):

$$t_{total} = t_{offline} + S \times \left(t_{online} + t_{postprocessing}\right) \tag{34}$$

with S being the number of DoE points considered – recall Eq. (3). This means that the composite toughness can be determined using SCA with 64 matrix clusters for a thousand DoE points ($S = 1000$) in a desktop computer in less than 8 h of computation (total).

On average, the direct numerical simulations of **each** high-fidelity RVE require 72 hours[7] of run time in a state of the art high performance computing cluster using 24 cores (Intel Haswell E5-2680v3 compute nodes with 2.5 GHz and 2×12-cores), and an additional 30 minutes to perform the homogenization of the stresses and strains using a single processor (postprocessing time). Clearly, the computational savings provided by the SCA method are a fundamental contribution to accomplish the data-driven design of the inelastic composite.

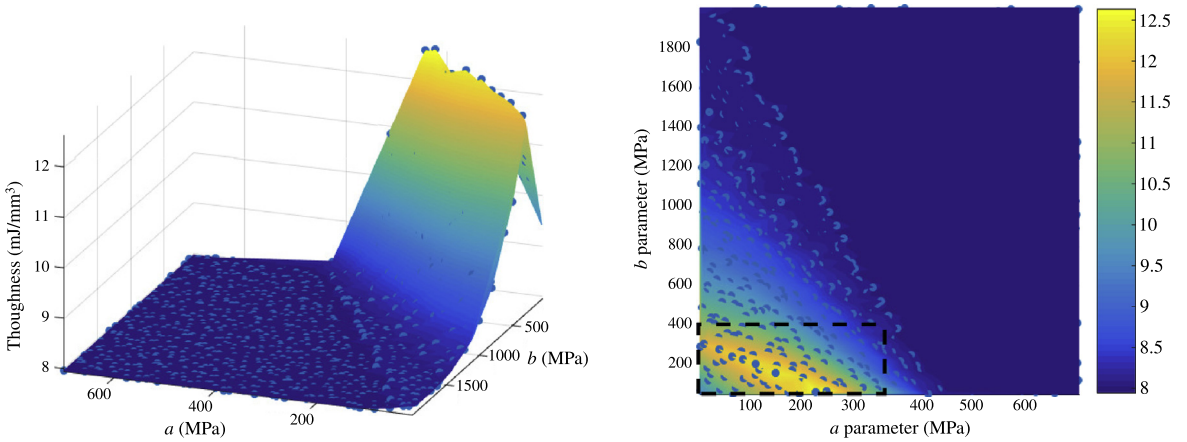
The DoE for the SCA analysis with 64 matrix clusters was obtained as described in the previous section and considering 1000 points from a Sobol sequence of the two input design variables with the following bounds:

$$a = [10,\ 700]\ \text{MPa}, \quad b = [50,\ 2000]\ \text{MPa}. \tag{35}$$

Fig. 13 shows a contour plot of the toughness property of the composite obtained for the 1000 DoE points within the above bounds of the input design variables. This figure shows a clear region where toughness is maximized (light yellow region) along a line with a small slope: for reference, the DoE points located close to that region have a toughness increase of 6.5% from $(a, b) \equiv (20,\ 310)$ MPa to $(a, b) \equiv (221,\ 60)$ MPa. The DoE point with highest predicted toughness from SCA considering $k_1 = 64$ matrix clusters occurs at $(a, b) \equiv (221, 60)$ MPa with a value of 12.6 mJ/mm^3.

[6] Computation times determined for a desktop computer using a single processor. Note that the SCA method is implemented in a non-optimized MATLAB® code.

[7] Note that the current problem requires a precise determination of the local strains and global stress/strain behavior of the composite in order to determine the fracture point. This implies a large number of analysis steps, so every simulation conducted in this section (whether for the SCA method or the DNS validation simulations) considered a total of 200 output steps for the uniaxial tension test of the composite up to a strain of 0.1.



(a) Response surface of composite toughness ($k_1 = 64$).

(b) Contour plot of composite toughness ($k_1 = 64$).

Fig. 13. Composite toughness obtained from the SCA method using 64 matrix clusters for 1000 DoE points of the inelastic parameters a and b. The dashed box indicates the region of interest where toughness is higher: $a = [100, 350]$ MPa and $b = [50, 400]$ MPa. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Comparison of the composite toughness obtained from the DNS of the high-fidelity RVE with the predictions obtained from SCA considering different number of material clusters. Value in parenthesis indicates relative error, toughness units are mJ/mm^3.

	DNS	SCA ($k_1 = 1024$)	SCA ($k_1 = 256$)	SCA ($k_1 = 64$)
$(a, b) \equiv (43, 1106)$	7.1	7.7 (+8.5%)	8.0 (+12.7%)	8.3 (+16.9%)
$(a, b) \equiv (221, 60)$	9.8	11.1 (+13.2%)	**12.0** (+22.4%)	**12.6** (+28.6%)
$(a, b) \equiv (91, 129)$	**11.4**	**11.4** (0.0%)	11.3 (−0.9%)	11.6 (+1.8%)

A qualitative analysis of the results in Fig. 13 suggests that the general trend of composite toughness variation with parameters a and b is captured. On the one hand for sufficiently high values of yielding (parameter a) the actual value of a becomes irrelevant because the particles remain linear elastic, i.e. composite fracture occurs by matrix cracking before the particles enter the inelastic regime, leading to the same value of toughness (dark blue region with 8 mJ/mm^3). On the other hand, even if the particles yield before the composite fractures (low values of a), if hardening is large enough (parameter b) then the composite toughness is still low. This leads to a contour plot where there are clear lines with similar levels of toughness – see Fig. 13(b).

A premature quantitative analysis of the results shown in Fig. 13 would indicate that points close to $(a, b) \equiv (221, 60)$ MPa would lead to maximum composite toughness. However, this response surface was obtained for a coarse analysis of a reduced RVE with 80 material clusters (64 for the matrix, 16 for the particles), as compared to 512,000 finite elements used in the DNS of the high-fidelity RVE.

In order to assess the accuracy of the SCA predictions, the stress–strain response of the composite for three specific DoE points is presented in Fig. 14. These points were selected according to Fig. 13(b), i.e. using the response surface obtained from a coarse SCA with $k_1 = 64$ clusters. In Fig. 14 there are three SCA predictions obtained for an increasing number of clusters, as well as the results obtained by DNS of the high-fidelity RVEs. The blue stress–strain curves are obtained for DoE point $(a, b) \equiv (43, 1106)$ MPa predicted to lead to low toughness since this point is far from the region of interest indicated by the box in Fig. 13. The other two points are located inside the region of interest corresponding to high toughness values. Table 5 summarizes the toughness predictions and compares the error of the various SCA refinements as compared to the DNS.

Observing Table 5 it can be seen that the composite toughness from DNS is higher for point $(a, b) \equiv (91, 129)$ than for point $(a, b) \equiv (221, 60)$, unlike what is predicted by the coarse SCA response ($k_1 = 64$). As the number of material clusters increases, one can observe from the table as well as Fig. 14 that the toughness approximation error decreases, and that using 1024 matrix clusters the SCA method correctly predicts which point has higher toughness.

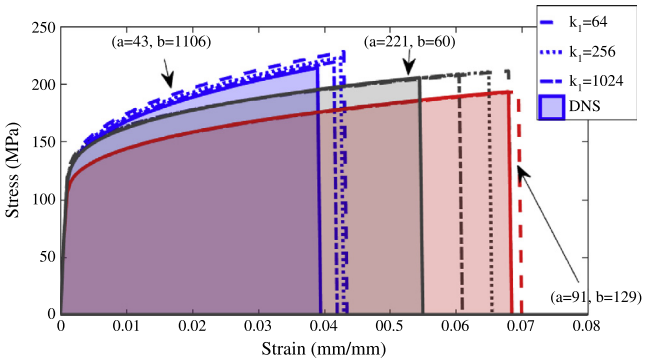


Fig. 14. Stress–strain response of the composite material for three different DoE points. The results obtained from the SCA method for different number of clusters and from the Direct Numerical Simulation (DNS) of the high-fidelity RVE are included for comparison. The composite toughness is represented as the shaded area underneath the curves. Table 5 includes the toughness values obtained for the three DoE points and respective SCA predictions.

Table 6
Comparison of computation times using a **single** core for SCA and DNS.

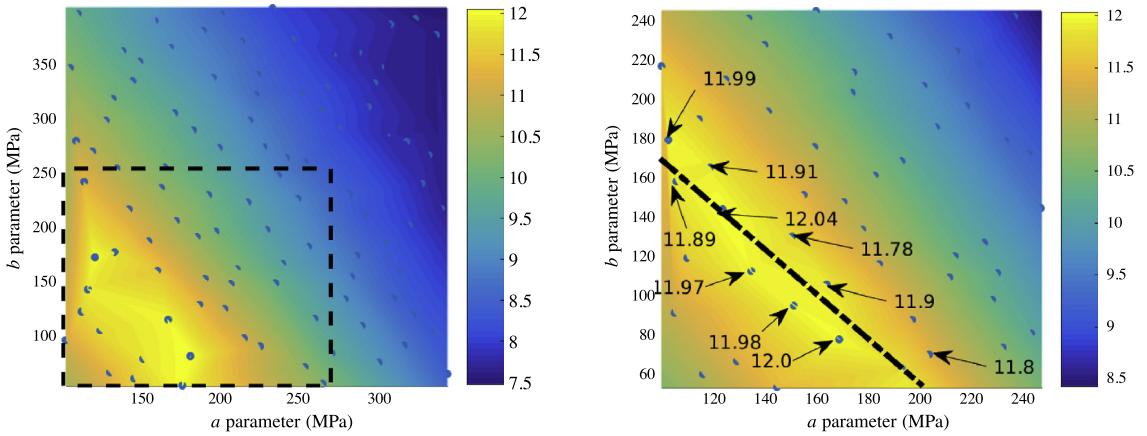
	DNS	SCA ($k_1 = 1024$)	SCA ($k_1 = 512$)	SCA ($k_1 = 64$)
Offline stage (**one time computation**)	0	31 h 20 min	6 h 27 min	10 min
Predictive stage (**per DoE point**)	≈1728 h	1 h 36 min	19 min	27 s
Postprocessing (**per DoE point**)	30 min	0	0	0

Interestingly, Table 5 demonstrates that the SCA method converges to the solution faster for some points than others. Recall that these analyses are particularly challenging because fracture depends on highly localized deformations. The SCA method was developed to be an optimal strategy to reproduce these local deformations using dramatically fewer discretization points while still capturing the global response as accurately as possible. Plasticity is typically a more diffuse form of deformation than fracture; therefore, a larger number of clusters is required for capturing damage. This is well illustrated in Fig. 14 where it is clear that the stress–strain curves are predicted very accurately but the onset of failure is less accurate. Nevertheless, the SCA method continues to converge to the solution as the number of material clusters increases while still showing a computation time that is several orders of magnitude lower – see Table 6.

Some notes to consider about the results in Table 6:

- The computation times for SCA are obtained using a non-optimized in-house MATLAB® code. The potential computational gains are significantly larger if using other programming language. Parallelization is also possible (the MATLAB® code is currently parallelized as well).
- Due to the large computation time needed for DNS, the time obtained for the predictive stage of the DNS is estimated by multiplying the actual simulation time by the number of cores used (24). This is a fair estimate since the parallelization occurs within a single compute node of the high performance computing cluster;
- the SCA method inherently includes the "postprocessing stage" because the system of Eqs. (B.2)–(B.3) includes the average stress or strain of the composite as unknowns or as constraints.

The SCA method allows for an adaptive refinement strategy of the design problem. The problem can be first approached by capturing the global trend of the response, as shown in Fig. 13. Then, choosing key DoE points we can evaluate the accuracy of the method by comparing the successive refinements of SCA predictions to the time consuming DNS (or even with experimental results), as done in Fig. 14. This validation enables the selection of an appropriate number of material clusters to use in SCA such that a smaller region of the design space can be re-sampled to find the local maxima. Fig. 15 illustrates this refinement process. Fig. 15(a) is a contour plot obtained with $k_1 = 512$ matrix clusters to determine the toughness for a DoE of 100 points in the region highlighted in Fig. 13(b). Note that only 67 points of the DoE used in Fig. 13(b) were within that region. Fig. 15(b) is a contour plot obtained with $k_1 = 1024$ for 50 DoE points generated within the domain highlighted in Fig. 15(a). In (a) there are 33 points within



(a) SCA with $k_1 = 512$. (b) SCA with $k_1 = 1024$.

Fig. 15. Composite toughness contour plot obtained from SCA with (a) 512 and (b) 1024 matrix clusters. The dashed box indicates the region of interest where toughness is higher: $a = [100, 250]$ MPa and $b = [50, 250]$ MPa.

the boxed region. As can be observed in Fig. 15 the successive refinement continues to show that there is a region along a line where the optimal toughness of the composite can be found. Successive refinements would lead to an even more accurate location of that optimal line. Also note that the design through the proposed data-driven framework leads to an approximate increase of 60% in toughness for the three-dimensional composite by tuning the particle constitutive law.

4. Conclusion

A new data-driven computational framework applicable to the design of structures and materials is developed. The synergistic choices of the design of experiments (DoE), computational analysis method (whether direct numerical simulations or a suitable reduced order model), and machine learning algorithm can assist in finding new structures, materials, properties and models.

Two illustrative examples for the framework are provided. In the first example the strategy of the framework is explained for a problem where the computational analyses and homogenization procedure are sufficiently fast to avoid the use of a reduced order model. This first example addressed several points:

1. Merits of using a non-uniform space filling design such as Sobol sequence for the DoE;
2. Considerations on the computational challenges involved in analyzing each DoE point, such as geometry generation, periodic boundary conditions and computational homogenization;
3. Influence of uncertainty at the input (imperfect descriptors) and output (imperfect predictive or experimental tools) level. For some problems, uncertainty quantification can be a topic of significant importance, especially when assessing the propagation of uncertainty from inputs to outputs;
4. Comparison of different machine learning algorithms.

The second illustrative example addressed a significantly more challenging problem: finding the influence of inelastic material phase parameters on the toughness of a three-dimensional composite material. The large computation times involved in analyzing this problem prevent the data-driven framework of using direct numerical simulations. A new numerical method previously developed by some authors of this article, self-consistent clustering analysis (SCA), is shown to be a viable solution to solve this challenge within a reasonable time frame.

As a final note, there is a vast number of opportunities for improvement of the data-driven framework presented herein. Integrating manufacturing variables in the design can be important for real applications. Selection of appropriate methods and/or development of new ones can also lead to tangible simplifications at each step of the framework:

1. Choice of the sampling strategy to perform the DoE can decrease the number of data points required to find the new design or model;
2. Different reduced order models can be of fundamental importance to form a reasonably sized database.
3. The choice of the machine learning algorithm can also be of significant importance. In some cases an appropriate choice can decrease the size of the database required.
4. A fourth research topic could be introduced in the data-driven framework presented herein: optimization [16,108]. This will be discussed in detail in a future publication.

The fourth point can be important in two different ways. Once a model is obtained through machine learning, optimization algorithms [108] can quickly find the global optimum of the model, which can be especially challenging for high-dimensional design spaces. Alternatively, optimization techniques such as genetic algorithms [108] may also provide a way to circumvent the need for machine learning by providing a direct method for sampling the design space and finding local optima without building a response surface model [16,19,20,93,94]. The latter approach, however, does not lead to a predictive model for untrained data which is useful for data-driven constitutive modeling as in the first example of this article.

Depending on the problem of interest, each step of the framework can assume different importance. Hopefully, this was effectively illustrated with the two examples explored herein.

Acknowledgment

The authors warmly thank the support from the Air Force Office of Scientific Research Grant No. FA9550-14-1-0032.

Appendix A. Computational analysis integration in data-driven design framework

Algorithm 1 outlines the framework, while Algorithm 2 outlines the respective function blocks. Comments are preceded by the ▷ symbol, files written for each analysis are denoted in *italic* with an arbitrary extension *.ext*, and blocks of code where different scripts and methods can be used are indicated by `<Insert block of code here>`. These algorithms are intended to guide the implementation process and to provide the necessary overview to understand the particular code developed.

Algorithms 1 and 2 were implemented in MATLAB®, which is only used for the purpose of writing the necessary files and calling the external software that executes these files automatically. The pre- and post-processing are performed by running the MATLAB® code which writes the python codes that are then executed in ABAQUS® pre- and post-processor, respectively. The finite element analyses files are also written from the MATLAB® code to external files that are then executed in ABAQUS® Explicit/Implicit. Only the reduced order model was fully implemented in MATLAB® including the computations, so there is no need to call any external software, as discussed in publication [46]. Note that the developed MATLAB® code calls all the external software automatically for all the points in the database, without the need for the analyst to intervene in the process manually.

Appendix B. An overview of the self-consistent clustering analysis

Recently, a new numerical method termed self-consistent clustering analysis (SCA) [46] was proposed by some of the authors of this article to accelerate the predictions of linear and nonlinear reversible and irreversible deformation of heterogeneous material RVEs. The idea is to reduce the computational cost of RVE analyses without compromising their high-fidelity by conjugating two efforts:

- decreasing the resolution of the numerical discretization (data compression);
- counterbalancing the loss in resolution with a more robust analysis method.

The SCA method is composed of two stages – offline and online stage – that can be summarized as follows. The offline stage consists of obtaining a reduced RVE where the domain is decomposed in a group of material clusters. Contrary to other reduced order models, SCA only requires linear elastic analyses of the high-fidelity RVE under three orthogonal loading conditions for two-dimensional RVEs, or six orthogonal loading conditions for three-dimensional ones.

Algorithm 1 Framework for database creation via integrated computational analyses

procedure CREATEDATABASE(DoE, QoI, $UseROM$, PBC, $ResponseDatabase$)
  ▷ DoE: M microstructures, R realizations, P property sets and L loadings
  ▷ QoI: array with quantities of interest (QoI) to homogenize
  ▷ $UseROM$: flag signaling that a Reduced Order Model (ROM) is being used
  ▷ PBC: flag signaling that Periodic Boundary Conditions (PBCs) are being used
  ▷ $ResponseDatabase$: output database with homogenized QoI
  Load DoE ▷ Load Design of Experiments structure variable
  for $n \leftarrow 1, N$ **do** ▷ Loop over N DoE points
    ▷ Extract geometry label m and realization label r for this DoE point n:
    $m \leftarrow DoE.labels(n, 1)$
    $r \leftarrow DoE.labels(n, 2)$
    $geometry \leftarrow DoE.geometry(m, r)$ ▷ Load geometry descriptors of this DoE point
    Call GETMESH($geometry$, PBC) ▷ Write FEA mesh file: *Mesh.ext*
    ▷ Extract property set label p for this DoE point n:
    $p \leftarrow DoE.labels(n, 3)$
    ▷ Load descriptors of property set p:
    $props \leftarrow DoE.props(p)$ ▷ Load properties of this DoE point
    if $UseROM = 1$ **then** ▷ If using a Reduced Order Model (ROM)
      ▷ Run offline stage of ROM:
      Call OFFLINEROM($props$) ▷ Get input file $ROMinput.ext$
    end if
    ▷ Extract loading label l for this DoE point n:
    $l \leftarrow DoE.labels(n, 4)$
    $load \leftarrow DoE.loads(l)$ ▷ Load loading of this DoE point
    if $UseROM = 1$ **then** ▷ If using a Reduced Order Model
      ▷ Run online stage of ROM:
      Call ONLINEROM($props$, $load$) ▷ Get output file $ROMoutput.ext$
    else ▷ If not using a Reduced Order Model
      ▷ Run nonlinear FEA:
      Call NONLINFEA($props$, $load$) ▷ Get output file $FEAoutput.ext$
    end if
    ▷ Homogenize all QoI for RVE (or reduced RVE) and update database
    Call HOMOGENIZERESULT(QoI, $ResponseDatabase$)
  end for
  return $ResponseDatabase$ ▷ Return database for subsequent machine learning
end procedure

The linear elastic analyses of the high-fidelity RVE are used to find a near-optimal domain decomposition by grouping points that have similar mechanical behavior under any applied boundary condition. Each group of points is called a material cluster and can be discontinuous. Material clusters are found by computing at every point of the high-fidelity RVE the strain concentration tensor $\mathbf{A}(\mathbf{x})$, and then using a pattern recognition algorithm called k-means clustering [109] to group the points with similar strain concentration tensors.

Since the strain concentration tensor $\mathbf{A}(\mathbf{x})$ is invariant in elasticity for any macroscopic deformation applied to the RVE due to the principle of superposition, an optimal domain decomposition of the RVE is determined, as illustrated in Fig. B.16. This figure presents three different reduced RVEs obtained from the same two-dimensional plane strain high-fidelity composite RVE with a mesh of 600×600 finite elements, as described in [46]. These reduced RVEs illustrate the refinement that can be achieved increasing the number of material clusters. For clarity the figure only shows the matrix phase (phase 1).

Once the k-means clustering domain decomposition is finished, the offline stage of SCA concludes by computing the interaction tensors $\mathbf{D}^{IJ}$ between every material cluster. These tensors represent the influence of the stress in the

Algorithm 2 Outline of function blocks used in Algorithm 1

function GETMESH($geometry$, PBC)
  ▷ Write file *PreScript.ext* with pre-processing script for this geometry:
  `<Write PreScript.ext>`
  Run *PreScript.ext* ▷ Get mesh file *Mesh.ext* by calling FEA pre-processing software
  if $PBC = 1$ **then** ▷ If using Periodic Boundary Conditions
    ▷ Update *Mesh.ext* file with periodic boundary condition constraints:
    `<Impose periodic boundary conditions>`
  end if
end function
function OFFLINEROM($props$)
  `<Run FEA analyses to train the chosen Reduced Order Model>`
  `<Model reduction applied to FEA analyses>` ▷ Offline or training stage
  ▷ This offline stage returns file *ROMinput.ext* with the ROM calibration database
end function
function ONLINEROM($props$, $load$)
  ▷ Run the ROM online stage for material property set $props$, boundary condition $load$, and using file *ROMinput.ext* that includes the ROM offline calibration database obtained for the realization r of the microstructure m:
  `<Online or predictive stage of the Reduced Order Model>`
  ▷ This procedure returns file *ROMoutput.ext* containing the local QoI outputs over the RVE domain Ω
end function
function NONLINFEA($props$, $load$)
  ▷ Write file with FEA conditions and local output variables. This file includes the mesh file *Mesh.ext*, uses the property set $props$ and boundary condition $load$:
  `<Write NonlinFEA.ext>`
  ▷ Conduct FEA by calling external software:
  Run *FEAinput.ext* ▷ Get output file: $FEAoutput.ext$
end function
function HOMOGENIZERESULT(QoI, $ResponseDatabase$)
  for all $q \in QoI$ **do** ▷ Loop over every requested Quantity of Interest in list QoI
    ▷ Numerically solve $\bar{q} = \frac{1}{\Omega}\int_\Omega q \, d\Omega$ over the RVE (or reduced RVE) domain Ω:
    `<Compute homogenized quantity of interest` $\bar{q}$`>`
    $ResponseDatabase \leftarrow \bar{q}$ ▷ Update response database
  end for
  ▷ Return database collecting every homogenized QoI:
  return $ResponseDatabase$
end function

Jth cluster on the strain in the Ith cluster. The interaction tensor $\mathbf{D}^{IJ}$ is written as an integral of Green's function in the high-fidelity RVE domain Ω with periodic boundary conditions,

$$\mathbf{D}^{IJ} = \frac{1}{c^I|\Omega|}\int_\Omega\int_\Omega \chi^I(\mathbf{x})\chi^J(\mathbf{x}')\boldsymbol{\Phi}^{\text{ref}}(\mathbf{x},\mathbf{x}')d\mathbf{x}'d\mathbf{x} \tag{B.1}$$

where c^I is the volume fraction of the Ith cluster, $|\Omega|$ is the volume of domain Ω, $\chi^I(\mathbf{x})$ is a window function, and $\boldsymbol{\Phi}^{\text{ref}}(\mathbf{x},\mathbf{x}')$ is the fourth-order periodic Green's function associated with an isotropic linear elastic reference material with stiffness tensor $\mathbf{C}^{\text{ref}}$. As shown in our work [46], this reference stiffness should be considered as the stiffness of the high-fidelity RVE to ensure the self-consistency of the method and improve convergence.

After computing the interaction tensors and finishing the offline stage, SCA can be used to predict the behavior of the reduced RVE – online or predictive stage. This stage can be performed for any boundary condition of choice and, more importantly, for any set of nonlinear constitutive laws with the same elastic properties without redoing the offline

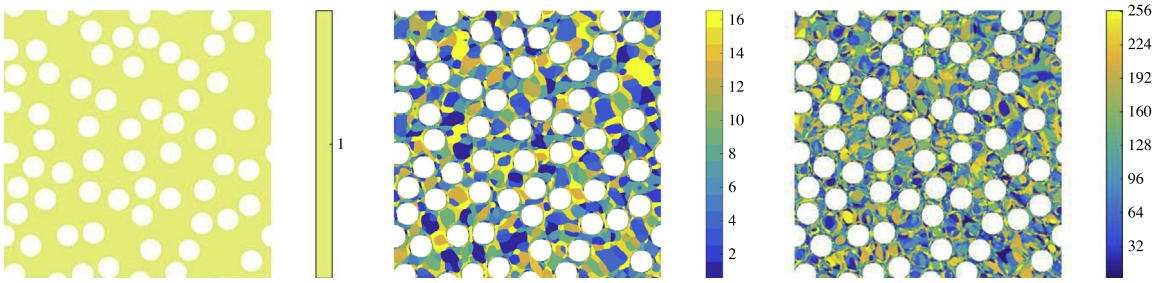


Fig. B.16. Three different reduced RVEs showing the subdomain decompositions of matrix phase (phase 1) obtained by **A**-based clustering. From left to right the number k_1 of clusters in the matrix phase is: $k_1 = 1$, $k_1 = 16$, and $k_1 = 256$.

stage. In other words, one can predict the plastic behavior of the RVE without conducting any of the computationally expensive plasticity analysis of the high-fidelity RVE.

The online stage of SCA consists of solving a system of equations obtained by averaging the Lippmann–Schwinger equation of each material cluster:

$$\Delta\boldsymbol{\varepsilon}^I + \sum_{J=1}^{k} \mathbf{D}^{IJ} : \left[\Delta\boldsymbol{\sigma}^J - \mathbf{C}^{\text{ref}} : \Delta\boldsymbol{\varepsilon}^J\right] - \Delta\boldsymbol{\varepsilon}^{\text{ref}} = 0 \tag{B.2}$$

from which a system of k equations is formed for all the clusters $I = 1, \ldots, k$ that is completed by considering the macro-strain or macro-stress constraints (boundary conditions) applied to the reduced RVE in incremental form:

$$\sum_{I=1}^{k} c^I \Delta\boldsymbol{\varepsilon}^I = \Delta\bar{\boldsymbol{\varepsilon}} \quad \text{or} \quad \sum_{I=1}^{k} c^I \Delta\boldsymbol{\sigma}^I = \Delta\bar{\boldsymbol{\sigma}} \tag{B.3}$$

where $\Delta\boldsymbol{\varepsilon}^J$ and $\Delta\boldsymbol{\sigma}^J$ denote the incremental stain and stress in the Jth cluster. Note that the stress in each cluster results from the local constitutive law of that cluster, i.e. if it is a cluster in the matrix phase the stress results from the plastic law of the matrix.

For details on the derivation of the above system of equations and on the numerical scheme the reader is referred to the original publication [46].

Appendix C. An overview of kriging and neural networks

C.1. Kriging

Kriging [92,110–112] is a nonlinear interpolation method that is based on a two-step process: (1) establishing a structure for the input design variables; and (2) interpolating the response obtained for each sample of the input design variables. The first step finds statistical relationships among the input design variables $\mathbf{x}$ by fitting a covariance and a degree of trend to them. The second step is the actual interpolation, similar to other methods.

Kriging is a general and statistically rigorous method for interpolating deterministic [92,110,111] or even stochastic [113] computer simulations. Non-parametric regression methods such as polynomial and spline interpolation do not include the first step which results in a loss of accuracy when compared to kriging [114,115]. In fact, spline interpolation can be shown to be equivalent to kriging with fixed covariance and degree of polynomial trend [114].

In order to simplify the notation and without loss of generality, a single quantity of interest $q \leftarrow q_i$ is assumed at this point. The basic idea of kriging is to approximate each scalar quantity of interest q as a realization of the random field $Q(\mathbf{x})$ over the space of input variables $\mathbb{R}^{d_{in}}$. If the random field $Q(\mathbf{x})$ is known, then it is possible to approximate the quantity of interest q ($\mathbf{x}^{new}$) at unsampled values in the database, i.e. for $\mathbf{x}^{new} \notin \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(S)}\}$ in Eq. (2).

In most kriging applications [92,112] it is common to restrict attention to linear predictors and to assume that the random field is a Gaussian process. An attractive feature of the kriging predictor is that it is the best linear unbiased predictor (BLUP) [114,115] in that it minimizes the mean square error (MSE) of prediction among all linear predictors. The Gaussian process $Q(\mathbf{x})$ is assumed to have parametric mean (or expected value),

$$E[Q(\mathbf{x})] = \mathbf{m}^T(\mathbf{x})\boldsymbol{\beta} \tag{C.1}$$

and covariance function,

$$c\left(\mathbf{x}^{(s)},\mathbf{x}^{(r)}\right)=Cov[Q(\mathbf{x}^{(s)}),Q(\mathbf{x}^{(r)})] \tag{C.2}$$

where $\mathbf{m}(\mathbf{x})=[m_1(\mathbf{x}),m_2(\mathbf{x}),\ldots m_u(\mathbf{x})]^T$ are u known basis functions (e.g. linear, quadratic, exponential, etc.), and $\boldsymbol{\beta}=[\beta_1,\beta_2,\ldots,\beta_u]^T$ is a vector of u unknown parameters. A common choice for the covariance function $c\left(\mathbf{x}^{(s)},\mathbf{x}^{(r)}\right)$ is the Gaussian covariance:

$$c\left(\mathbf{x}^{(s)},\mathbf{x}^{(r)}\right)=\sigma^2\exp\left[\sum_{j=1}^{d_{in}}-w_j(x_j^{(r)}-x_j^{(s)})^2\right] \tag{C.3}$$

where σ^2 is the prior variance of the Gaussian process $Q(\mathbf{x})$ and $\mathbf{w}=[w_1,w_2,\ldots,w_{d_{in}}]^T$ are the correlation (roughness) parameters which control the smoothness of the random field (a large w_j is an indication of a rough response surface along dimension j).

The variance σ^2 as well as vectors $\boldsymbol{\beta}$ and $\mathbf{w}$ is unknown and need to be estimated, respectively $\hat{\sigma}^2$, $\hat{\boldsymbol{\beta}}$, and $\hat{\mathbf{w}}$. This estimation can be done by different methods such as maximum likelihood estimation or cross-validation [112]. The maximum likelihood estimation is equivalent to maximizing the logarithm of the likelihood function,

$$[\hat{\boldsymbol{\beta}},\hat{\sigma}^2,\hat{\mathbf{w}}]=\underset{\boldsymbol{\beta},\sigma^2,\mathbf{w}}{\operatorname{argmax}}\left\{\log\left(\mathcal{L}\left[\boldsymbol{\beta},\sigma^2,\mathbf{w}|\mathbf{q}\right]\right)\right\} \tag{C.4}$$

which is commonly considered to be the multivariate Gaussian likelihood function,

$$[\hat{\boldsymbol{\beta}},\hat{\sigma}^2,\hat{\mathbf{w}}]=\underset{\boldsymbol{\beta},\sigma^2,\mathbf{w}}{\operatorname{argmax}}\left\{\log\left(\frac{1}{\sigma^S\sqrt{2\pi|\mathbf{C}|}}\exp\left[-\frac{1}{2\sigma^2}(\mathbf{q}-\mathbf{M}\boldsymbol{\beta})^T\mathbf{C}^{-1}(\mathbf{q}-\mathbf{M}\boldsymbol{\beta})\right]\right)\right\} \tag{C.5}$$

where $\mathbf{q}=[q\left(\mathbf{x}^{(1)}\right),q\left(\mathbf{x}^{(2)}\right),\ldots,q\left(\mathbf{x}^{(S)}\right)]^T$ is the $S\times 1$ vector with the response quantity of interest for the S samples of the input variables, $\mathbf{M}$ is an $S\times u$ matrix with sth row of $\mathbf{m}^T\left(\mathbf{x}^{(s)}\right)$, and $\mathbf{C}$ is an $S\times S$ matrix with each element (r,s) given by $c\left(\mathbf{x}^{(r)},\mathbf{x}^{(s)}\right)$.

A possible approach for maximizing Eq. (C.5) is to represent $\boldsymbol{\beta}$ and σ^2 as a function of $\mathbf{w}$ and then performing the maximization by setting the partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{\beta}$ and σ^2 to zero, yielding:

$$\hat{\boldsymbol{\beta}}=\left(\mathbf{M}^T\mathbf{C}^{-1}\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{C}^{-1} \tag{C.6}$$

$$\hat{\sigma}^2=\frac{1}{S}\left(\mathbf{q}-\mathbf{M}\hat{\boldsymbol{\beta}}\right)^T\mathbf{C}^{-1}\left(\mathbf{q}-\mathbf{M}\hat{\boldsymbol{\beta}}\right). \tag{C.7}$$

Noting that Eqs. (C.6) and (C.7) only depend on $\mathbf{w}$ (since $\mathbf{C}$ depends on $\mathbf{w}$), replacing these results in Eq. (C.5) and performing the maximization leads to a prediction for $\hat{\mathbf{w}}$. After obtaining $\hat{\mathbf{w}}$, parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are found from Eqs. (C.6) and (C.7), respectively. This method is known in the literature as profiling [116].

Once the parameters $\hat{\mathbf{w}}$, $\hat{\boldsymbol{\beta}}$, and $\hat{\sigma}^2$ are estimated, the predicted estimate $\hat{q}\left(\mathbf{x}^{new}\right)$ of the response $q\left(\mathbf{x}^{new}\right)$ at unsampled input points $\mathbf{x}^{new}$ is given by:

$$\hat{q}\left(\mathbf{x}^{new}\right)=\mathbf{m}^T\left(\mathbf{x}^{new}\right)\hat{\boldsymbol{\beta}}+\mathbf{c}^T\left(\mathbf{x}^{new}\right)\mathbf{C}^{-1}\left(\mathbf{q}-\mathbf{M}\hat{\boldsymbol{\beta}}\right) \tag{C.8}$$

and the associated mean square error (MSE) of the prediction is

$$MSE\left[\hat{q}\left(\mathbf{x}^{new}\right)\right]=c\left(\mathbf{x}^{new},\mathbf{x}^{new}\right)-\mathbf{c}^T\left(\mathbf{x}^{new}\right)\mathbf{C}^{-1}\mathbf{c}\left(\mathbf{x}^{new}\right)+\mathbf{W}^T\left(\mathbf{M}^T\mathbf{C}^{-1}\mathbf{M}\right)^{-1}\mathbf{W} \tag{C.9}$$

where $\mathbf{W}=\mathbf{m}^T\left(\mathbf{x}^{new}\right)-\mathbf{M}^T\mathbf{C}^{-1}\mathbf{c}\left(\mathbf{x}^{new}\right)$ and $\mathbf{c}\left(\mathbf{x}^{new}\right)$ is an $S\times 1$ vector whose sth element is $c\left(\mathbf{x}^{(s)},\mathbf{x}^{new}\right)$. As a final note, since kriging is being used herein for interpolation, reversion to the mean [117,118] is of no concern; hence, constant prior mean (i.e., $m(\mathbf{x})=1$ and $u=1$) is used henceforth.

If multiple quantities of interest are defined instead of a single quantity of interest $q\left(\mathbf{x}^{(s)}\right)$, then the above equations are computed d_{out} times in order to obtain each $q_i\left(\mathbf{x}^{(s)}\right)$, see Eq. (3). Alternatively, a multiresponse Gaussian Process model may be used [119].

The benefits of kriging are the ability to quantify the prediction uncertainty and to handle highly nonlinear behavior. However, kriging is applicable to a relatively small number of samples due to the computational costs associated with the inversion of the covariance matrix.

C.2. Neural networks

Neural networks [120–124] assume that each output quantity of interest q_i results from applying a chosen transformation function f to a quantity called neuron n_i,

$$q_i = f(n_i) \tag{C.10}$$

where each neuron n_i is just a linear combination of all the input variables x_j,

$$n_i = w_{ij} x_j + b_i \tag{C.11}$$

and where w_{ij} are the weights to be determined, b_i the bias or offset parameters that are also unknown, and f the transfer or activation function chosen by the analyst. Note that Einstein's summation convention is adopted henceforth ($w_{ij}x_j = \sum_j w_{ij}x_j$).

The choice of the transfer function f depends on the machine learning problem to be solved, as discussed by Hagan et al. [124]. Among many possibilities there are three classical examples that can be invoked: the hard limit function, the linear transformation function, and the log-sigmoid function. The hard limit function is a step function[8] that is suitable for classification problems: when the neuron is below a certain value the output is classified in one category, otherwise it is classified in another. The linear transformation function consists of having the output variable to be the same as the neuron $q_i = f(n_i) = n_i$. The log-sigmoid function is defined as

$$f(n_i) = \frac{1}{1 + e^{-n_i}} \tag{C.12}$$

and is particularly common when the neural network is used to do metamodeling or function approximation, as in the problems considered in this article. This transfer function is differentiable, which is useful for finding the weights w_{ij} and biases b_i via the most common iterative schemes.

Neural networks can have multiple layers. These multilayer networks are the result of an assembly of neurons into a specific network architecture. This is possible because neurons can act on the outputs of a previous neuron,

$$n_i^{\lambda+1} = w_{ij}^{\lambda+1} q_j^{\lambda} + b_i^{\lambda+1}, \text{ for } \lambda = 0, 1, \ldots, \Lambda - 1 \tag{C.13}$$

with $\lambda + 1$ being the respective layer and Λ the total number of layers in the network. Then, each neuron corresponds to an output variable after applying the transfer function,

$$q_i^{\lambda+1} = f(n_i^{\lambda+1}) = f(w_{ij}^{\lambda+1} q_j^{\lambda} + b_i^{\lambda+1}) \,, \text{ for } \lambda = 0, 1, \ldots, \Lambda - 1 \tag{C.14}$$

where the design inputs x_j intuitively correspond to the initial variables q_j^0,

$$q_j^0 = x_j \tag{C.15}$$

and where the last output variables q_i^{Λ} correspond to the design quantities of interest q_i,

$$q_i^{\Lambda} = q_i. \tag{C.16}$$

The special case when neurons act directly on the problem design inputs x_i to get the output design quantities of interest q_i occurs for a single layer network ($\Lambda = 1$). The remaining cases require the analyst to choose an architecture for the multilayer network, the simplest and most common of which being the feedforward architecture. As the name suggests, in this architecture the design inputs are propagated forward through the successive neurons until reaching the design outputs via Eq. (C.14).

Besides choosing the network architecture the analyst also needs to decide on how to train the neural network from a given database. Training a neural network is the process of determining the weights $w_{ij}^{\lambda+1}$ and biases $b_i^{\lambda+1}$ for every neuron in each layer. Several iterative schemes have been proposed [124], being the most common the backpropagation algorithm proposed by Werbos [125].

[8] The output being 1 when the input is greater than 0, and 0 otherwise.

In the backpropagation scheme the training data consisting of S sets of inputs x_j and outputs q_i, see (3), are compared with the predicted outputs $\hat{q}_i$ obtained at each iteration of the training stage. This comparison is called performance index and is equivalent to minimize the mean square error:

$$E(w_{ij}^{\lambda+1}, b_i^{\lambda+1}) = (q_i - \hat{q}_i)(q_i - \hat{q}_i) \tag{C.17}$$

where the dependence of the performance index E on all the weights $w_{ij}^{\lambda+1}$ and biases $b_i^{\lambda+1}$ is explicitly written.

Finding the weights and biases in the backpropagation algorithm is achieved by minimizing E through successive iterations using the steepest descent algorithm:

$$w_{ij}^{\lambda}|_{k+1} = w_{ij}^{\lambda}|_k - \gamma \left.\frac{\partial E}{\partial w_{ij}^{\lambda}}\right|_k \tag{C.18}$$

$$b_i^{\lambda}|_{k+1} = b_i^{\lambda}|_k - \gamma \left.\frac{\partial E}{\partial b_i^{\lambda}}\right|_k \tag{C.19}$$

where $|_k$ denotes iteration k, and γ is the step size of the steepest descent algorithm that is allowed to change in each iteration. Computing the derivatives using the chain rule leads to:

$$\frac{\partial E}{\partial w_{ij}^{\lambda}} = \frac{\partial E}{\partial n_r^{\lambda}} \frac{\partial n_r^{\lambda}}{\partial w_{ij}^{\lambda}} = \frac{\partial E}{\partial n_i^{\lambda}} q_j^{\lambda-1} \tag{C.20}$$

$$\frac{\partial E}{\partial b_i^{\lambda}} = \frac{\partial E}{\partial n_r^{\lambda}} \frac{\partial n_r^{\lambda}}{\partial b_i^{\lambda+1}} = \frac{\partial E}{\partial n_i^{\lambda}} \tag{C.21}$$

where the partial derivatives $\frac{\partial n_r^{\lambda}}{\partial w_{ij}^{\lambda}} = \delta_{ri} q_j^{\lambda}$ and $\frac{\partial n_r^{\lambda}}{\partial b_i^{\lambda}} = \delta_{ri}$ follow from Eq. (C.13).

Since two consecutive neurons are closely related, the term $\frac{\partial E}{\partial n_i^{\lambda}}$ can be calculated using the chain rule again:

$$\frac{\partial E}{\partial n_i^{\lambda}} = \frac{\partial E}{\partial n_r^{\lambda+1}} \frac{\partial n_r^{\lambda+1}}{\partial n_i^{\lambda}} \tag{C.22}$$

where the last term is calculated from Eqs. (C.13) and (C.14):

$$\frac{\partial n_r^{\lambda+1}}{\partial n_i^{\lambda}} = w_{rp}^{\lambda+1} \frac{\partial q_p^{\lambda}}{\partial n_i^{\lambda}} = w_{rp}^{\lambda+1} \frac{\partial f\left(n_p^{\lambda}\right)}{\partial n_i^{\lambda}} = \sum_p w_{rp}^{\lambda+1} \frac{\partial f\left(n_p^{\lambda}\right)}{\partial n_p^{\lambda}} \tag{C.23}$$

since $\frac{\partial f\left(n_p^{\lambda}\right)}{\partial n_i^{\lambda}} = 0$ for $p \neq i$, and noting that the summation sign is explicitly written on the RHS of the above equation because the index p is repeated three times.

Replacing Eq. (C.23) in (C.22),

$$\frac{\partial E}{\partial n_i^{\lambda}} = \frac{\partial E}{\partial n_r^{\lambda+1}} \left(\sum_p w_{rp}^{\lambda+1} \frac{\partial f\left(n_p^{\lambda}\right)}{\partial n_p^{\lambda}} \right) \tag{C.24}$$

a recurrence relation is obtained where $\frac{\partial E}{\partial n_i^{\lambda}}$ is determined from the next value, i.e. the derivatives are propagated backwards (hence the name backpropagation algorithm). This recurrence relation needs to be initiated by defining the derivative of the performance index at the last layer Λ. Since the performance index was defined as depending on the predicted value $\hat{q}_i$, the following result for the last layer is achieved:

$$\frac{\partial E}{\partial n_i^{\Lambda}} = \frac{\partial \left[\left(q_j - \hat{q}_j\right)\left(q_j - \hat{q}_j\right)\right]}{\partial n_i^{\Lambda}} = -2\left(q_i - \hat{q}_i\right) \frac{\partial f\left(n_i^{\Lambda}\right)}{\partial n_i^{\Lambda}} \quad \text{(no sum on } i\text{)}. \tag{C.25}$$

This completes the backpropagation algorithm. In summary, in neural networks with a feedforward architecture and the backpropagation algorithm the information starts by traveling forward from the inputs x_j to the outputs q_i:

$$q_j^0 = x_j \tag{C.26}$$

$$q_i^{\lambda+1} = f(w_{ij}^{\lambda+1} q_j^{\lambda} + b_i^{\lambda+1}), \text{ for } \lambda = 0, 1, \ldots, \Lambda - 1 \tag{C.27}$$

$$q_i^{\Lambda} = q_i \tag{C.28}$$

then, the derivatives of the mean least square error are computed starting on the last layer:

$$\frac{\partial E}{\partial n_i^{\Lambda}} = -2\left(q_i - \hat{q}_i\right)\frac{\partial f^{\lambda}\left(n_i^{\Lambda}\right)}{\partial n_i^{\Lambda}} \quad \text{(no sum on } i\text{)} \tag{C.29}$$

and moving backwards until the first layer in a recurrent relation:

$$\frac{\partial E}{\partial n_i^{\lambda}} = \frac{\partial E}{\partial n_r^{\lambda+1}}\left(\sum_p w_{rp}^{\lambda+1}\frac{\partial f^{\lambda}\left(n_p^{\lambda}\right)}{\partial n_p^{\lambda}}\right), \text{ for } \lambda = \Lambda - 1, \ldots, 2, 1 \tag{C.30}$$

from which the weights and biases can be finally estimated for the iterative step $k+1$ by the steepest descent algorithm from Eqs. (C.18) and (C.19):

$$w_{ij}^{\lambda}|_{k+1} = w_{ij}^{\lambda}|_k - \gamma\left.\frac{\partial E}{\partial n_i^{\lambda}}\right|_k q_j^{\lambda-1} \tag{C.31}$$

$$b_i^{\lambda}|_{k+1} = b_i^{\lambda}|_k - \gamma\left.\frac{\partial E}{\partial n_i^{\lambda}}\right|_k \tag{C.32}$$

Appendix D. Summary of global sensitivity analysis

The main sensitivity index is written as,

$$S_i = \frac{V_i}{V(q)} \tag{D.1}$$

where $V(q)$ is the total unconditional variance defined as a summation of the partial variances [126],

$$V(q) = \sum_i^{d_{in}} V_i + \sum_i^{d_{in}}\sum_{j>i}^{d_{in}} V_{ij} + \cdots \tag{D.2}$$

with d_{in} being the total number of input variables, $\sum_i^{d_{in}} V_i$ the sum of partial variances that include the first-order effects of each single input variable, $\sum_i^{d_{in}}\sum_{j>i}^{d_{in}} V_{ij}$ the sum of partial variances that include the interaction of two input variables, etc.

The total sensitivity index is written as,

$$S_{T_i} = S_i + \sum_{i\neq j} S_{ij} + \sum_{i\neq j\neq k} S_{ijk} + \cdots \tag{D.3}$$

with the number of indices $i, j, k, \ldots$ being limited by the total number of input variables. The higher-order sensitivity indices are given by,

$$S_{ij\ldots d_{in}} = \frac{V_{ij\ldots d_{in}}}{V(q)}. \tag{D.4}$$

Different variance estimations have been proposed by different authors, as reviewed by Saltelli et al. [107]. A common formulation when using Sobol sequence leads to the following result [107]:

$$S_i = \frac{\frac{1}{N_s}\sum_{(m,p,l)=1}^{N_s}\bar{q}_{\mathbf{B}}^{(m,p,l)}\left(\bar{q}_{\mathbf{A}_{\mathbf{B}}^i}^{(m,p,l)} - \bar{q}_{\mathbf{A}}^{(m,p,l)}\right)}{\frac{1}{N_s}\sum_{(m,p,l)=1}^{N_s}\left(\bar{q}_{\mathbf{A}}^{(m,p,l)} - \bar{\bar{q}}_{\mathbf{A}}^{(m,p,l)}\right)^2} \tag{D.5}$$

$$S_{T_i} = \frac{\frac{1}{2N_s}\sum_{(m,p,l)=1}^{N_s}\left(\bar{q}_{\mathbf{A}}^{(m,p,l)} - \bar{q}_{\mathbf{A}_{\mathbf{B}}^i}^{(m,p,l)}\right)^2}{\frac{1}{N_s}\sum_{(m,p,l)=1}^{N_s}\left(\bar{q}_{\mathbf{A}}^{(m,p,l)} - \bar{\bar{q}}_{\mathbf{A}}^{(m,p,l)}\right)^2} \tag{D.6}$$

where $\bar{\bar{q}}_{\mathbf{A}}^{(m,p,l)}$ is given by,

$$\bar{\bar{q}}_{\mathbf{A}}^{(m,p,l)} = \frac{1}{N_s} \sum_{(m,p,l)=1}^{N_s} \bar{q}_{\mathbf{A}}^{(m,p,l)} \tag{D.7}$$

and where $\bar{q}_{\mathbf{A}}^{(m,p,l)}$ is the model output $\bar{q}$ obtained for DoE point (m, p, l) included in a subset $\mathbf{A}$ with N_s points of the DoE for all the input variables, $\bar{q}_{\mathbf{B}}^{(m,p,l)}$ is the output for another subset $\mathbf{B}$ of the DoE with N_s points, and $\bar{q}_{\mathbf{A}_\mathbf{B}^i}^{(m,p,l)}$ is the response for the subset $\mathbf{A}$ but where the values of the input variable x_i are replaced by the values of that variable in subset $\mathbf{B}$.

References

[1] M.A. Meyers, P.-Y. Chen, A.Y.-M. Lin, Y. Seki, Biological materials: Structure and mechanical properties, Prog. Mater. Sci. (ISSN: 0079-6425) 53 (1) (2008) 1–206.

[2] D. Jang, L.R. Meza, F. Greer, J.R. Greer, Fabrication and deformation of three-dimensional hollow ceramic nanostructures, Nat. Mater. 12 (10) (2013) 893–898.

[3] P. Wang, F. Casadei, S. Shan, J.C. Weaver, K. Bertoldi, Harnessing buckling to design tunable locally resonant acoustic metamaterials, Phys. Rev. Lett. 113 (2014) 014301.

[4] J. Fröhlich, W. Niedermeier, H.-D. Luginsland, The effect of filler–filler and filler–elastomer interaction on rubber reinforcement, Composites A (ISSN: 1359-835X) 36 (4) (2005) 449–460.

[5] G. Heinrich, M. Klüppel, T.A. Vilgis, Reinforcement of elastomers, Curr. Opin. Solid State Mater. Sci. (ISSN: 1359-0286) 6 (3) (2002) 195–203.

[6] F. Hussain, M. Hojjati, M. Okamoto, R.E. Gorga, Review article: Polymer-matrix nanocomposites, processing, manufacturing, and application: An overview, J. Compos. Mater. 40 (17) (2006) 1511–1575.

[7] L.J. Lee, C. Zeng, X. Cao, X. Han, J. Shen, G. Xu, Polymer nanocomposite foams, Compos. Sci. Technol. (ISSN: 0266-3538) 65 (15–16) (2005) 2344–2363.

[8] S. Tjong, Novel nanoparticle-reinforced metal matrix composites with enhanced mechanical properties, Adv. Energy Mater. (ISSN: 1527-2648) 9 (8) (2007) 639–652.

[9] Y. Swolfs, L. Gorbatikh, I. Verpoest, Fibre hybridisation in polymer composites: A review, Composites A (ISSN: 1359-835X) 67 (2014) 181–200.

[10] R. Tavares, A. Melro, M.A. Bessa, A. Turon, W.K. Liu, P. Camanho, Mechanics of hybrid polymer composites: analytical and computational study, Comput. Mech. 57 (3) (2016) 405–421.

[11] J. Yvonnet, D. Gonzalez, Q.-C. He, Numerically explicit potentials for the homogenization of nonlinear elastic heterogeneous materials, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 198 (33–36) (2009) 2723–2737.

[12] A. Clément, C. Soize, J. Yvonnet, Computational nonlinear stochastic homogenization using a nonconcurrent multiscale approach for hyperelastic heterogeneous microstructures analysis, Internat. J. Numer. Methods Engrg. (ISSN: 1097-0207) 91 (8) (2012) 799–824.

[13] J. Yvonnet, E. Monteiro, Q.-C. He, Computational homogenization method and reduced database model for hyperelastic heterogeneous structures, Int. J. Multiscale Comput. Eng. (ISSN: 1543-1649) 11 (3) (2013) 201–225.

[14] B. Le, J. Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks, Internat. J. Numer. Methods Engrg. 104 (12) (2015) 1061–1084.

[15] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 304 (2016) 81–101.

[16] O. Sigmund, Materials with prescribed constitutive parameters: An inverse homogenization problem, Int. J. Solids Struct. (ISSN: 0020-7683) 31 (17) (1994) 2313–2329.

[17] O. Sigmund, A 99 line topology optimization code written in Matlab, Struct. Multidiscip. Optim. (ISSN: 1615-1488) 21 (2) (2001) 120–127.

[18] M.P. Bendsoe, O. Sigmund, Topology Optimization: Theory, Methods, and Applications, Springer Science & Business Media, 2013.

[19] Z. Gürdal, R.T. Haftka, P. Hajela, Design and Optimization of Laminated Composite Materials, John Wiley & Sons, 1999.

[20] K. Deb, S. Gulati, Design of truss-structures for minimum weight using genetic algorithms, Finite Elem. Anal. Des. (ISSN: 0168-874X) 37 (5) (2001) 447–465 Genetic algorithms and finite elements in engineering.

[21] X. Ning, S. Pellegrino, Imperfection-insensitive axially loaded thin cylindrical shells, Int. J. Solids Struct. (ISSN: 0020-7683) 62 (2015) 39–51.

[22] H. Xu, Y. Li, C. Brinson, W. Chen, A descriptor-based design methodology for developing heterogeneous microstructural materials system, J. Mech. Des. (ISSN: 1050-0472) 136 (5) (2014) 051007.

[23] H. Xu, R. Liu, A. Choudhary, W. Chen, A machine learning-based design representation method for designing heterogeneous microstructures, J. Mech. Des. (ISSN: 1050-0472) 137 (5) (2015) 051403.

[24] Y. Zhang, H. Zhao, I. Hassinger, L.C. Brinson, L.S. Schadler, W. Chen, Microstructure reconstruction and structural equation modeling for computational design of nanodielectrics, Integ. Mater. Manuf. Innov. (ISSN: 2193-9772) 4 (1) (2015) 14.

[25] R. Bostanabad, A.T. Bui, W. Xie, D.W. Apley, W. Chen, Stochastic microstructure characterization and reconstruction via supervised learning, Acta Mater. (ISSN: 1359-6454) 103 (2016) 89–102.

[26] T.W. Simpson, D.K. Lin, W. Chen, Sampling strategies for computer experiments: design and analysis, Int. J. Reliab. Appl. 2 (3) (2001) 209–240.

[27] K.-T. Fang, R. Li, A. Sudjianto, Design and Modeling for Computer Experiments, CRC Press, 2005.
[28] A. Jourdan, J. Franco, Optimal Latin hypercube designs for the Kullback–Leibler criterion, AStA Adv. Stat. Anal. 94 (4) (2010) 341–351.
[29] J. Santiago, M. Claeys-Bruno, M. Sergent, Construction of space-filling designs using {WSP} algorithm for high dimensional spaces, Chemometr. Intell. Lab. Syst. (ISSN: 0169-7439) 113 (2012) 26–31.
[30] H. Faure, Discrépance de suites associées à un système de numération (en dimension s), Acta Arith. 41 (4) (1982) 337–351.
[31] J.M. Hammersley, Monte Carlo methods for solving multivariable problems, Ann. New York Acad. Sci. 86 (3) (1960) 844–874.
[32] I.M. Sobol, On the distribution of points in a cube and the approximate evaluation of integrals, Zh. Vychisl. Mat. Mat. Fiz. 7 (4) (1967) 784–802.
[33] I.M. Sobol, Uniformly distributed sequences with an additional uniform property, USSR Comput. Math. Math. Phys. 16 (5) (1976) 236–242.
[34] R. Bates, R. Buck, E. Riccomagno, H. Wynn, Experimental design and observation for large systems, J. R. Stat. Soc. Ser. B Stat. Methodol. (1996) 77–94.
[35] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[36] R. Jin, W. Chen, A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments, J. Statist. Plann. Inference (ISSN: 0378-3758) 134 (1) (2005) 268–287.
[37] A.B. Owen, Orthogonal arrays for computer experiments, integration and visualization, Statist. Sinica (1992) 439–452.
[38] M. Petelet, B. Iooss, O. Asserin, A. Loredo, Latin hypercube sampling with inequality constraints, AStA Adv. Stat. Anal. (ISSN: 1863-818X) 94 (4) (2010) 325–339.
[39] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, Internat. J. Numer. Methods Engrg. (ISSN: 1097-0207) 37 (2) (1994) 229–256.
[40] M. Bessa, J. Foster, T. Belytschko, W. Liu, A meshfree unification: reproducing kernel peridynamics, Comput. Mech. (ISSN: 0178-7675) 53 (6) (2014) 1251–1264.
[41] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Internat. J. Numer. Methods Fluids (ISSN: 1097-0363) 20 (8–9) (1995) 1081–1106.
[42] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 194 (39–41) (2005) 4135–4195.
[43] A. Melro, P. Camanho, F.A. Pires, S. Pinho, Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part {II} – Micromechanical analyses, Int. J. Solids Struct. (ISSN: 0020-7683) 50 (11–12) (2013) 1906–1915.
[44] X. Bai, M. Bessa, A. Melro, P. Camanho, L. Guo, W. Liu, High-fidelity micro-scale modeling of the thermo-visco-plastic behavior of carbon fiber polymer matrix composites, Compos. Struct. (ISSN: 0263-8223) (2015).
[45] M.A. Bessa, Data-Driven Multi-Scale Analyses of Materials and Structures, (Ph.D. thesis), Northwestern University, 2016.
[46] Z. Liu, M. Bessa, W.K. Liu, Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 306 (2016) 319–341.
[47] J.A. Moore, R. Ma, A.G. Domel, W.K. Liu, An efficient multiscale model of damping properties for filled elastomers with complex microstructures, Composites B (ISSN: 1359-8368) 62 (2014) 262–270.
[48] Z. Liu, J.A. Moore, W.K. Liu, An extended micromechanics method for probing interphase properties in polymer nanocomposites, J. Mech. Phys. Solids (ISSN: 0022-5096) 95 (2016) 663–680.
[49] G.J. Dvorak, Transformation field analysis of inelastic composite materials, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 437 (1900) (1992) 311–327.
[50] J. Michel, P. Suquet, Nonuniform transformation field analysis, Int. J. Solids Struct. (ISSN: 0020-7683) 40 (25) (2003) 6937–6955.
[51] K. Karhunen, Zur spektraltheorie stochastischer prozesse, Suomalainen tiedeakatemia, 1946.
[52] M. Loève, Probability Theory; Foundations, Random Sequences, D. Van Nostrand Company, New York, 1955.
[53] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2002.
[54] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1) (1993) 539–575.
[55] J. Yvonnet, Q.-C. He, The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains, J. Comput. Phys. 223 (1) (2007) 341–368.
[56] P. Ladevèze, J.-C. Passieux, D. Néron, The {LATIN} multiscale computational method and the Proper Generalized Decomposition, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 199 (21–22) (2010) 1287–1296.
[57] F. Chinesta, A. Ammar, A. Leygue, R. Keunings, An overview of the proper generalized decomposition with applications in computational rheology, J. Non-Newton. Fluid Mech. (ISSN: 0377-0257) 166 (11) (2011) 578–592.
[58] J. Chaboche, P. Kanouté, A. Roos, On the capabilities of mean-field approaches for the description of plasticity in metal matrix composites, Int. J. Plast. (ISSN: 0749-6419) 21 (7) (2005) 1409–1434.
[59] C. Oskay, J. Fish, Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 196 (7) (2007) 1216–1243.
[60] J. Fish, Practical Multiscaling, John Wiley & Sons, 2013.
[61] P. Krysl, S. Lall, J. Marsden, Dimensional model reduction in non-linear finite element dynamics of solids and structures, Internat. J. Numer. Methods Engrg. 51 (4) (2001) 479–504.
[62] 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. (ISSN: 1432-0924) 58 (2) (2016) 213–234.

[63] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, A. Huerta, PGD-Based computational vademecum for efficient design, optimization and control, Arch. Comput. Methods Eng. (ISSN: 1886-1784) 20 (1) (2013) 31–59.
[64] T. Reichhardt, et al., It's sink or swim as a tidal wave of data approaches, Nature 399 (6736) (1999) 517–520.
[65] C. Lynch, Big data: How do your data grow?, Nature 455 (7209) (2008) 28–29.
[66] W. Los, J. Wood, Dealing with data: Upgrading infrastructure, Science (ISSN: 0036-8075) 331 (6024) (2011) 1515–1516.
[67] C.A. Mattmann, Computing: A vision for data science, Nature 493 (7433) (2013) 473–475.
[68] T.M. Mitchell, Mining our reality, Science (ISSN: 0036-8075) 326 (5960) (2009) 1644–1645.
[69] R. Sachidanandam, D. Weissman, S.C. Schmidt, J.M. Kakol, L.D. Stein, G. Marth, S. Sherry, J.C. Mullikin, B.J. Mortimore, D.L. Willey, et al., A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms, Nature 409 (6822) (2001) 928–933.
[70] H. Akil, M.E. Martone, D.C. Van Essen, Challenges and opportunities in mining neuroscience data, Science (ISSN: 0036-8075) 331 (6018) (2011) 708–712.
[71] R. Bellazzi, B. Zupan, Predictive data mining in clinical medicine: current issues and guidelines, International journal of medical informatics 77 (2) (2008) 81–97.
[72] S. Buonamici, T. Trimarchi, M.G. Ruocco, L. Reavie, S. Cathelin, B.G. Mar, A. Klinakis, Y. Lukyanov, J.-C. Tseng, F. Sen, et al., CCR7 signalling as an essential regulator of CNS infiltration in T-cell leukaemia, Nature 459 (7249) (2009) 1000–1004.
[73] S.M. Hanash, S.J. Pitteri, V.M. Faca, Mining the plasma proteome for cancer biomarkers, Nature 452 (7187) (2008) 571–579.
[74] L.I. Shlush, S. Zandi, A. Mitchell, W.C. Chen, J.M. Brandwein, V. Gupta, J.A. Kennedy, A.D. Schimmer, A.C. Schuh, K.W. Yee, et al., Identification of pre-leukaemic haematopoietic stem cells in acute leukaemia, Nature 506 (7488) (2014) 328–333.
[75] J.-H. Hehemann, G. Correc, T. Barbeyron, W. Helbert, M. Czjzek, G. Michel, Transfer of carbohydrate-active enzymes from marine bacteria to Japanese gut microbiota, Nature 464 (7290) (2010) 908–912.
[76] F. Warnecke, P. Luginbühl, N. Ivanova, M. Ghassemian, T.H. Richardson, J.T. Stege, M. Cayouette, A.C. McHardy, G. Djordjevic, N. Aboushadi, et al., Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite, Nature 450 (7169) (2007) 560–565.
[77] A. Vinayagam, Y. Hu, M. Kulkarni, C. Roesel, R. Sopko, S.E. Mohr, N. Perrimon, Protein complex–Based Analysis Framework for High-Throughput Data Sets, Science Signaling (ISSN: 1945-0877) 6 (264) (2013) rs5.
[78] J. Besnard, G.F. Ruda, V. Setola, K. Abecassis, R.M. Rodriguiz, X.-P. Huang, S. Norval, M.F. Sassano, A.I. Shin, L.A. Webster, et al., Automated design of ligands to polypharmacological profiles, Nature 492 (7428) (2012) 215–220.
[79] N.P. Tatonetti, P.P. Ye, R. Daneshjou, R.B. Altman, Data-driven prediction of drug effects and interactions, Sci. Transl. Med. (ISSN: 1946-6234) 4 (125) (2012) 125ra31.
[80] A. Cully, J. Clune, D. Tarapore, J.-B. Mouret, Robots that can adapt like animals, Nature 521 (7553) (2015) 503–507.
[81] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[82] R. Whelan, R. Watts, C.A. Orr, R.R. Althoff, E. Artiges, T. Banaschewski, G.J. Barker, A.L. Bokde, C. Büchel, F.M. Carvalho, et al., Neuropsychosocial profiles of current and future adolescent alcohol misusers, Nature 512 (7513) (2014) 185–189.
[83] E. Chavez, G. Conway, M. Ghil, M. Sadler, An end-to-end assessment of extreme weather impacts on food security, Nature Clim. Change (ISSN: 1758-678X) 5 (11) (2015) 997–1001.
[84] C.E. Yoon, O. O'Reilly, K.J. Bergen, G.C. Beroza, Earthquake detection through computationally efficient similarity search, Sci. Adv. (ISSN: 2375-2548) 1 (11) (2015) e1501057.
[85] L. Einav, J. Levin, Economics in the age of big data, Science (ISSN: 0036-8075) 346 (6210) (2014).
[86] S. Curtarolo, D. Morgan, K. Persson, J. Rodgers, G. Ceder, Predicting crystal structures with data mining of quantum calculations, Phys. Rev. Lett. 91 (2003) 135503.
[87] C.C. Fischer, K.J. Tibbetts, D. Morgan, G. Ceder, Predicting crystal structure by merging data mining with quantum mechanics, Nat. Mater. 5 (8) (2006) 641–646.
[88] S. Curtarolo, G.L. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nat. Mater. 12 (3) (2013) 191–201.
[89] P. Raccuglia, K.C. Elbert, P.D. Adler, C. Falk, M.B. Wenny, A. Mollo, M. Zeller, S.A. Friedler, J. Schrier, A.J. Norquist, Machine-learning-assisted materials discovery using failed experiments, Nature 533 (7601) (2016) 73–76.
[90] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R.H. Taylor, L.J. Nelson, G.L. Hart, S. Sanvito, M. Buongiorno-Nardelli, et al., AFLOWLIB. ORG: A distributed materials properties repository from high-throughput ab initio calculations, Comput. Mater. Sci. 58 (2012) 227–235.
[91] J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD), Jom 65 (11) (2013) 1501–1509.
[92] T. Simpson, J. Poplinski, N.P. Koch, J. Allen, Metamodels for computer-based engineering design: Survey and recommendations, Eng. Comput. (ISSN: 1435-5663) 17 (2) (2001) 129–150.
[93] J. Schutte, A. Groenwold, Sizing design of truss structures using particle swarms, Struct. Multidiscip. Optim. (ISSN: 1615-1488) 25 (4) (2003) 261–269.
[94] M. Sonmez, Discrete optimum design of truss structures using artificial bee colony algorithm, Struct. Multidiscip. Optim. (ISSN: 1615-1488) 43 (1) (2011) 85–97.
[95] E.M. Arruda, M.C. Boyce, A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials, J. Mech. Phys. Solids 41 (2) (1993) 389–412.

[96] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, Nonlinear Fnite Eements for Continua and Structures, John Wiley & Sons, 2013.
[97] F. Feyel, J.-L. Chaboche, FE 2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials, Comput. Methods Appl. Mech. Engrg. 183 (3) (2000) 309–330.
[98] V. Kouznetsova, M.G.D. Geers, W.A.M. Brekelmans, Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme, Internat. J. Numer. Methods Engrg. (ISSN: 1097-0207) 54 (8) (2002) 1235–1260.
[99] F. Feyel, A multilevel finite element method (FE2) to describe the response of highly non-linear structures using generalized continua, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 192 (28–30) (2003) 3233–3244.
[100] J. Fish, S. Kuznetsov, Computational continua, Internat. J. Numer. Methods Engrg. (ISSN: 1097-0207) 84 (7) (2010) 774–802.
[101] S. Ghosh, K. Lee, S. Moorthy, Two scale analysis of heterogeneous elastic-plastic materials with asymptotic homogenization and voronoi cell finite element model, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 132 (1–2) (1996) 63–116.
[102] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 157 (1–2) (1998) 69–94.
[103] K. Terada, M. Hori, T. Kyoya, N. Kikuchi, Simulation of the multi-scale convergence in computational homogenization approaches, Int. J. Solids Struct. (ISSN: 0020-7683) 37 (16) (2000) 2285–2311.
[104] A. Melro, P. Camanho, S. Pinho, Generation of random distribution of fibres in long-fibre reinforced composites, Compos. Sci. Technol. (ISSN: 0266-3538) 68 (9) (2008) 2092–2102.
[105] A. Melro, P. Camanho, F.A. Pires, S. Pinho, Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I –Constitutive modelling, Int. J. Solids Struct. (ISSN: 0020-7683) 50 (11–12) (2013) 1897–1905.
[106] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, John Wiley & Sons, 2008.
[107] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index, Comput. Phys. Comm. 181 (2) (2010) 259–270.
[108] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. (ISSN: 1089-778X) 6 (2) (2002) 182–197.
[109] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, 1967, pp. 281–297.
[110] D.G. Krige, A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand, (Ph.D. thesis), University of Witwatersrand, 1951.
[111] G. Matheron, Principles of geostatistics, Econom. Geol. 58 (8) (1963) 1246–1266.
[112] S. Banerjee, B.P. Carlin, A.E. Gelfand, Hierarchical Modeling and Analysis for Spatial Data, Crc Press, 2014.
[113] B. Ankenman, B.L. Nelson, J. Staum, Stochastic kriging for simulation metamodeling, Oper. Res. 58 (2) (2010) 371–382.
[114] O. Dubrule, Comparing splines and kriging, Comput. Geosci. (ISSN: 0098-3004) 10 (2) (1984) 327–338.
[115] G.M. Laslett, Kriging and splines: An empirical comparison of their predictive performance in some applications, J. Amer. Statist. Assoc. 89 (426) (1994) 391–400.
[116] P. McCullagh, J.A. Nelder, Generalized Linear Models, Vol. 37, CRC press, 1989.
[117] V.R. Joseph, Limit kriging, Technometrics 48 (4) (2006) 458–466.
[118] J. Staum, Better simulation metamodeling: The why, what, and how of stochastic kriging, in: Proceedings of the 2009 Winter Simulation Conference, WSC, ISSN: 0891-7736, 2009, pp. 119–133.
[119] S. Conti, A. O'Hagan, Bayesian emulation of complex multi-output and dynamic computer models, J. Statist. Plann. Inference 140 (3) (2010) 640–651.
[120] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386.
[121] B. Widrow, M.E. Hoff, Adaptive switching circuits, IRE WESCON Convention Record (ISSN: 2375-2548) 4 (August 1960) 96–104.
[122] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (8) (1982) 2554–2558.
[123] D.E. Rumelhart, J.L. McClelland, CORPORATE PDP Research Group (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, MIT Press, Cambridge, MA, USA, ISBN: 0-262-68053-X, 1986.
[124] M.T. Hagan, H.B. Demuth, M.H. Beale, O. De Jesús, Neural Network Design, 1996.
[125] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, (Ph.D. thesis), Harvard University, Cambridge, MA, 1974.
[126] I.M. Sobol, On sensitivity estimation for nonlinear mathematical models, Mat. Model. 2 (1) (1990) 112–118.