Computational Mechanics
https://doi.org/10.1007/s00466-024-02588-9

Computational homogenization for aerogel-like polydisperse open-porous materials using neural network-based surrogate models on the microscale

Axel Klawonn[1,2] · Martin Lanser[1,2] · Lucas Mager[1,3] · Ameya Rege[3,4]

Received: 1 March 2024 / Accepted: 19 December 2024

Abstract

The morphology of nanostructured materials exhibiting a polydisperse porous space, such as aerogels, is very open porous and fine grained. Therefore, a simulation of the deformation of a large aerogel structure resolving the nanostructure would be extremely expensive. Thus, multi-scale or homogenization approaches have to be considered. Here, a computational scale bridging approach based on the FE^2 method is suggested, where the macroscopic scale is discretized using finite elements while the microstructure of the open-porous material is resolved as a network of Euler–Bernoulli beams. Here, the beam frame based RVEs (representative volume elements) have pores whose size distribution follows the measured values for a specific material. This is a well-known approach to model aerogel structures. For the computational homogenization, an approach to average the first Piola–Kirchhoff stresses in a beam frame by neglecting rotational moments is suggested. To further overcome the computationally most expensive part in the homogenization method, that is, solving the RVEs and averaging their stress fields, a surrogate model is introduced based on neural networks. The network's input is the localized deformation gradient on the macroscopic scale and its output is the averaged stress for the specific material. It is trained on data generated by the beam frame based approach. The effiency and robustness of both homogenization approaches is shown numerically, the approximation properties of the surrogate model is verified for different macroscopic problems and discretizations. Different (Quasi-)Newton solvers are considered on the macroscopic scale and compared with respect to their convergence properties.

Keywords Open-porous material · Polydispersity · Aerogel · Homogenization · FE^2 · Finite elements · Beam frame · Neural networks · Machine learning

1 Introduction

The mechanics of open-porous materials form a fascinating branch of materials science, unveiling a complex interplay of structural intricacies and mechanical behaviors that have profound implications across multiple disciplines. These materials, characterized by their interconnected network of voids or pores, exhibit unique mechanical properties driven by their open-porous architecture [1]. The mechanical performance is influenced by factors such as pore size, shape, distribution, and the material composition itself. Understanding the deformation mechanisms and stress distribution within these porous structures is crucial for optimizing their performance in applications ranging from lightweight structural compo-

Dedicated to Jörg Schröder on the occasion of his 60th birthday.

✉ Axel Klawonn
axel.klawonn@uni-koeln.de

✉ Martin Lanser
martin.lanser@uni-koeln.de

✉ Lucas Mager
lucas.mager@uni-koeln.de

✉ Ameya Rege
ameya.rege@utwente.nl

[1] Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany

[2] Center for Data and Simulation Science, University of Cologne, Albertus-Magnus-Platz, 50923 Cologne, Germany

[3] Institute of Materials Research, German Aerospace Center, Linder Höhe, 51147 Cologne, Germany

[4] Department of Mechanics of Solids, Surfaces and Systems, University of Twente, PO Box 217, 7500 AE Enschede, Netherlands

Published online: 01 February 2025

nents to advanced filtration systems. A special class of such materials is the territory of nanoporous materials, such as nanofoams and aerogels. Aerogels exhibit pore sizes ranging from as low as less than two nanometers to as high as over hundreds of nanometers within the same material [2]. Such a polydisperse nature significantly affects the thermal conductivity and sound absorption characteristics, but also their overall mechanical performance [3]. Formulating an intricate understanding of the macroscopic mechanics of such polydisperse open-porous materials demands the theoretical understanding of the underlying deformation mechanisms. While state-of-the-art experimental tools help to dive deep into the microstructure of the material, computational modeling has proven to establish concrete structure–property relations providing, generally, a physics-informed explanation.

Diverse studies have reported on the computational modeling of open-porous materials that exhibit pores on the macroscale, however, very few studies have investigated the computational description of nanostructured open-porous materials. Here, we focus and limit our search to physics-informed or micromechanical models. Existing literature uses either constitutive modeling or computational homogenization as a means to describe the mechanical behavior of open-porous materials. Such materials are made up of a network of three-dimensionally interconnected struts or pore-walls. One of the first reports on the mathematical description of the deformation in such materials was presented by Gent and Thomas [4], where they described the network of elastic foams to be formed of dumbbell-shaped elements called threads and dead volumes. Only extension of the threads, namely axial deformation in the network was considered. A much more robust theory was presented later by Gibson and Ashby, who modeled these struts as Euler–Bernoulli beams and provided a description of their bulk mechanical behavior [5, 6]. While there were extensions of these models presented, a notable development was presented by Warren and Kraynik [7] who modeled a perfectly ordered foam, whose unit cell was inspired by a regular tetrakaidecahedron referred to as the open-foam Kelvin cell. This accounted for a more complex geometrical representation of a unit cell as compared to the previously considered square or hexagonal cells in two dimensions, or cubic cells in three dimensions. This model was inspired by the work of Dement'ev and Tarakanov [8], who studied plastic foams. While the Kelvin-cell model in [7] investigated only the elastic properties, the approach was extended to describe large deformations and crushing of foams, particularly metal foams using the finite-element-method [9, 10]. Interestingly, the model by Dement'ev and Tarakanov could show very good validations under large deformations with elastic foams. On the similar grounds of the strain energy approach, first proposed in [8], Rege et al. [11] proposed a generalized micromechanical constitutive model, based on the representation of cell walls as beams, to describe the macroscopic stress–strain response of aerogel-like open-porous materials. The model was extended to capture the densification in a later study and showed very good validation with biopolymer aerogels [12]. These above-mentioned models, both constitutive or homogenization ones, base the models on a representative unit cell, strictly speaking one that represents 'a pore' and the surrounding pore walls. This is however insufficient to describe more polydisperse open-porous materials. To this end, the construction of a representative volume element (RVE) is necessary that accounts for the complete pore space and the diversity in structural features of the material in consideration. In such a case, the Voronoi-tessellation-approach is widely used. In such a representation, the network is made up of Voronoi cells, and the cell walls, representing struts, are modeled again as beams. Analysis of such a modeling approach in application to open-porous materials has been reported by several authors [13–17]. These studies have however mostly dealt with two-dimensional problems. For three-dimensional modeling, most reports have used computer tomography images and reconstructed them for structure–property analysis [18–21]. While this provides a realistic picture of the microstructure, this approach becomes increasingly challenging as the pore sizes become smaller, particularly, when one is dealing with those below 50 nm, which is the case, for example in aerogels. Chandrasekaran et al. [22] proposed a radical Voronoi method, wherein the pore space was represented by a random closed pack of polydisperse spheres and the Laguerre-Voronoi-tessellations were generated over the spheres. A heuristic analysis of the microstructural parameters was subsequently analyzed by Aney and Rege [23]. However, this approach of analyzing the RVE only helps understand the bulk behavior under some classical loading scenarios and does not account for geometrical effects that may arise due to the micro–macro transition. It is here where a multiscale homogenization approach is demanded and lacking in the literature for such highly open-porous polydisperse materials. This becomes increasingly important for large heterogeneous RVEs because resolving their structure on the macroscopic scale using, for example, a beam model results in large linear systems which cannot be solved without enormous computational effort. To overcome this limitation, we suggest to use a computational homogenization approach based on the well-known FE^2 method [24–28].

In the FE^2 approach, the microstructure of the considered material is only resolved locally on the second scale, usually named microscopic scale. In this article, instead of using finite element discretizations on both levels, we modify the FE^2 method and use beam frame models to resolve the aerogel structure on the mesoscale. This is a natural choice, since the use of beams to model aerogel structures on the mesoscopic scale is a state-of-the-art approach and already used and suggested in [22, 29–31]. Also in the field of lattice structures the analysis of microscopic beam networks is an important research topic [32]. Multiscale methods [33–35] as well as beam theory-based [36–39] and machine learning [40] approaches are used for the analysis of these materials. Honeycomb lattice structures are of special interest for this research since it is a quite common structural element in nature and engineering. However, methods for lattice structures usually cannot be applied in the case of aerogels due to their unstructured arrangement of the fibrils. In the recent years data-driven computing [41, 42] has increased in popularity and has also found application in the simulation of porous materials [43] including structure–property predictions of aerogels [44, 45].

In general, the FE^2 method is computationally demanding and to increase the computational efficiency and reduce the time to solution, parallel implementations have been used a lot in the past, cf. [46–48]. An alternative to the extensive use of parallel computing is to drastically reduce the computational complexity of the FE^2 approach by training and exploiting machine learning based surrogate models on the microscale. Combinations of surrogate models based on deep neural networks (DNNs) and the FE^2 method are often denoted by DNN-FE^2 approaches and for different finite element applications, such methods have already been successfully studied in recent years. They have been proven to be robust and computationally very efficient in different contexts. Some of these approaches can be found in [49–59]. Based on these existing DNN-FE^2 methods, we suggest a neural network (NN) based model which is applied locally in all integration points of the macroscopic finite element discretization to predict the average stress depending on the local macroscopic deformation. Therefore, the DNN replaces the solution of local beam frame problems and the averaging of stresses in the usual homogenization approach. In contrast to, e.g., [49–60], we consider porous aerogel-like structures on the microscale and we use beam frame models to create the data to train the DNN surrogate model.

To the best of our knowledge, in the present work, we extend the large family of already existing homogenization approaches with an FE^2-related method based on beam frame models on the microscale and, for the first time, train deep learning-based surrogate models for this method. We also, for the first time, extensively discuss the use of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [61] as macroscopic nonlinear solver in FE^2 for porous materials, which is a good alternative if the consistent tangent is not available or expensive to use.

The remainder of the article is organized as follows. In Sect. 2 we describe the computational homogenization approach using beam frame models on the microscale. We replace the beam frame models by a faster surrogate model based on NNs in Sect. 3. Finally, in Sect. 4, we show numerical results and discuss the convergence of the different methods and the quality and accuracy of the results.

2 Computational homogenization methods

Performing macroscopic simulations resolving the microscopic morphology of aerogels and similar porous materials on a single scale is computationally extremely expensive. Therefore, we consider a computational scale bridging approach where the morphological representation of the open-porous material in consideration is resolved in localized representative volume elements (RVEs). Our approach is based on the well-known FE^2 method [24–28]. This method has been developed for the simulation of micro-heterogeneous solid materials as, for example, dual- or multi-phase steels. Usually, the macroscopic problem is discretized with comparably large finite elements while the finite element discretization of the RVEs resolves the microscopic structure. In our deviation from the FE^2 method, we also use a finite element discretization on the macroscopic scale but a beam frame model on the microscopic scale which suits the nanostructure of the aerogel. Following the FE^2 framework, in each Gaussian integration point of the macroscopic problem, one RVE of the aerogel structure discretized with beams is attached. Then, the RVE is deformed with respect to the local macroscopic deformation in the corresponding integration point and delivers, as a response, the macroscopic stress in that point by averaging over the stresses in the beam elements of the RVE. More details on the modeling of both scales are given in the following sections.

2.1 Modeling the nanostructure of the RVE

The nanostructure of a typical open-porous material can be seen in Fig. 1 which shows the scanning electron microscope (SEM) image of a carrageenan aerogel. For a two-scale homogenization approach, an RVE that is representative of the material's morphology, namely one that represents the entire pore space of the material in consideration, is desired. The RVE which represents such a nanostructure is generated using a method presented in [17, 22] for the case of biopolymer aerogels. The pore size distribution and the pore-wall diameter are required as an input for the generation algorithm. These parameters are obtained by experimental analysis such as SEM images and the Barrett-Joyner-Halenda (BJH) analysis of the physisorption isotherms and may vary depending on the exact procedure of the synthesis of the aerogel. A detailed description of the experimental analysis of the material can be found in [30].

The first step of the RVE generation method is to use a sphere packing algorithm where the sphere volume distribu-

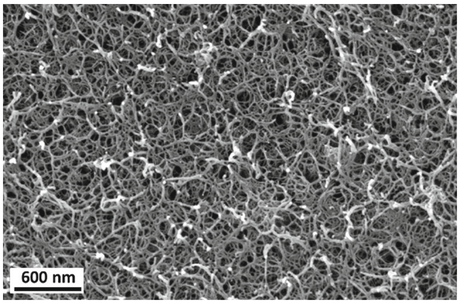


Fig. 1 An SEM image of a biopolymer aerogel. The figure is adapted from Rege et al. [30]

tion coalesces with the experimentally measured pore volume distribution of the given open-porous material. Afterwards, the sphere centers and the corresponding diameters are used in a Laguerre–Voronoi tessellation and the interfaces of the resulting Voronoi cells finally build the structure of the RVE. The struts on the interfaces represent the pore walls of the open-porous material. For a two-dimensional RVE the modeling is illustrated in Fig. 2. For more details, see also [17, 22].

To ensure periodic boundaries in each dimension it is possible to copy the spheres obtained from the sphere packing algorithm in each dimension and thus expand the domain. Cutting out the center of the Laguerre–Voronoi tessellation resulting from the increased number of sphere centers yields a periodic domain of the same size as the original domain and with continuous pore walls on each of the periodic boundaries. A visualization of this approach is presented in Fig. 3 where the green squares mark the copied domains and the red square highlights the periodic boundaries of the center domain. To enable the use of periodic boundary conditions on the microscopic scale of our FE^2 method we use this approach for the generation of the RVEs.

We also like to mention that due to the random nature of the generation method there is always a risk that the created RVE contains orthotropy. However, testing the RVEs used in this work has shown that in the regarded cases orthotropy has only been small and did not influence the computed solutions significantly. More detailed investigation of the the methods influence on orthotropy in the generated RVEs will not be part of this paper but could be an interesting topic for some future work.

2.2 Microscopic problem

On the microscopic level, the deformation of the pore walls in the RVEs is computed in each step of the overall homogenization algorithm and the resulting stress tensors are averaged afterwards. In the RVE, all the pore walls are modeled as Euler–Bernoulli beams and in the literature the resulting model is often related to as beam frame model [62].

We consider an RVE consisting of n_B beam elements and n_N nodes which are all connected by the beams. The computation of the deformation and rotation of each of the beam vertices on the microscopic scale requires solving a linear system of equations

$$K \cdot u = F.$$

Here, the stiffness matrix K is assembled from beam stiffness matrices K_e for each beam element e [63]. We consider a single beam element $e = \left[v_i, v_j\right]$ with a circular cross section and with starting point and endpoint

$$v_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \text{ and } v_j = \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix}.$$

Then, the beam stiffness matrix is given by

$$K_e = \mathbf{T}^T \cdot \frac{1}{L_e} \begin{pmatrix} \mathbf{D} & \mathbf{C}^T & -\mathbf{D} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{R_1} & \mathbf{C}^T & \mathbf{R_2} \\ -\mathbf{D} & \mathbf{C} & \mathbf{D} & \mathbf{C} \\ \mathbf{C} & \mathbf{R_2} & \mathbf{C}^T & \mathbf{R_1} \end{pmatrix} \cdot \mathbf{T},$$

where the submatrices $\mathbf{D}$, $\mathbf{R_1}$, and $\mathbf{R_2}$ are related to the deformation and the rotation of the beam element. The matrix $\mathbf{C}$ refers to the coupling of these variables. For the three-dimensional case the submatrices are defined as

$$\mathbf{D} = \begin{pmatrix} E \cdot A & 0 & 0 \\ 0 & 12\frac{E \cdot I}{L_e^2} & 0 \\ 0 & 0 & 12\frac{E \cdot I}{L_e^2} \end{pmatrix},$$

$$\mathbf{R}_1 = \begin{pmatrix} \frac{E \cdot I^2}{2 \cdot (1+\nu)} & 0 & 0 \\ 0 & 4E \cdot I & 0 \\ 0 & 0 & 4E \cdot I \end{pmatrix},$$

$$\mathbf{R}_2 = \begin{pmatrix} -\frac{E \cdot I^2}{2 \cdot (1+\nu)} & 0 & 0 \\ 0 & 2E \cdot I & 0 \\ 0 & 0 & 2E \cdot I \end{pmatrix},$$

$$\text{and } \mathbf{C} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -6\frac{E \cdot I}{L_e} \\ 0 & 6\frac{E \cdot I}{L_e} & 0 \end{pmatrix}.$$

Finally, $\mathbf{T}$ is a transformation matrix which depends on the orientation of the beam in space and the parameters E, I, A, and L_e refer to Young's modulus, the second moment of area, the area of the cross section, and the length of the beam. For a detailed description of the beam frame model and the Euler–Bernoulli beam theory, we refer to [63].

The systems right hand side $F = (\mathbf{Q}_1, \mathbf{M}_1, \mathbf{Q}_2, \mathbf{M}_2 \ldots, \mathbf{Q}_{n_N}, \mathbf{M}_{n_N})^T$ consists of forces $\mathbf{Q}_i = \left(Q_i^x, Q_i^y, Q_i^z\right)^T$ and bending moments $\mathbf{M}_i = \left(M_i^x, M_i^y, M_i^z\right)^T$ at each

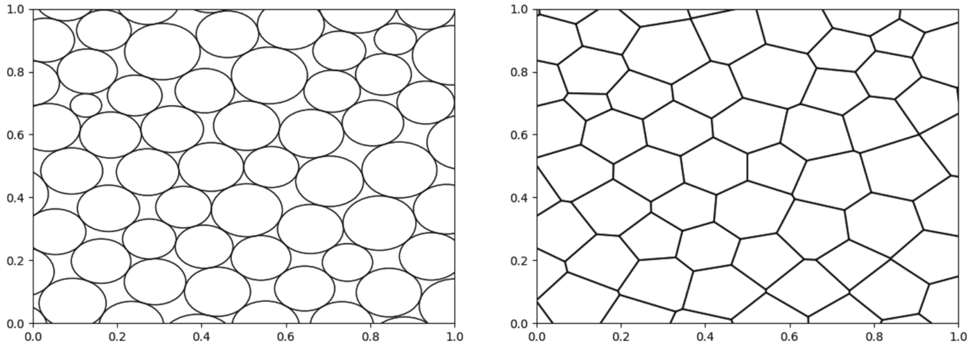


Fig. 2 Circle packing (left) and Voronoi tessellation (right) for the generation of the two-dimensional RVE

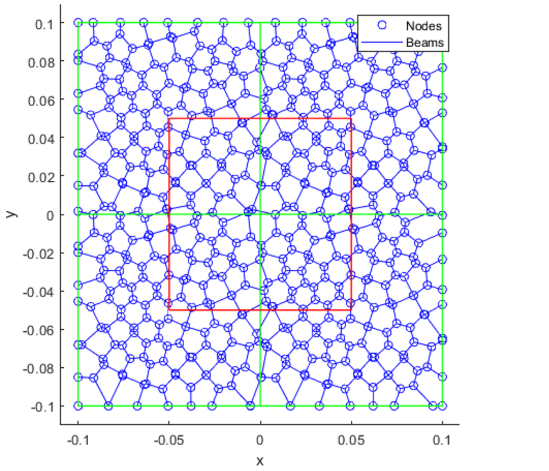


Fig. 3 Voronoi tessellation that results from copied sphere centers. Red square in the middle has periodic boundaries

node. Solving the equation yields the solution vector $u = \left(\mathbf{u}_1, \boldsymbol{\theta}_1, \mathbf{u}_2, \boldsymbol{\theta}_2, \ldots, \mathbf{u}_{n_N}, \boldsymbol{\theta}_{n_N}\right)^T$, where $\mathbf{u}_i = \left(u_i^x, u_i^y, u_i^z\right)^T$ refers to the deformation of each node and $\boldsymbol{\theta}_i = \left(\theta_i^x, \theta_i^y, \theta_i^z\right)^T$ is the vector of the corresponding rotations in each direction. By applying the beam frame model as described, it is assumed that the deformation within one beam element can be described by a cubic polynomial depending on its distance from the starting point of the beam. Let us introduce the variable $\xi \in [0, 1]$ as a local measurement for each beam for the normalized distance from each of the ends. For the beam element $e = [v_i, v_j]$ the value $\xi = 0$ for example refers to the global coordinates $v_i = (x_i, y_i, z_i)^T$ and the value $\xi = 1$ refers to the other node $v_j = \left(x_j, y_j, z_j\right)^T$. With this local variable it is possible to describe the deformation of each beam over its length in terms of ξ as $u^{x_k}(x, y, z) = \tilde{u}^{x_k}(\xi)$ for each $x_k = (x, y, z)$, where $\tilde{u}$ is a polynomial of third degree.

In general, the deformation of each node v_i, $i = 1, \ldots, n_N$, in the RVE can be split into two parts

$$\mathbf{u}_i = \tilde{\mathbf{u}}_i + \bar{\mathbf{u}}_i.$$

Here, $\bar{\mathbf{u}}_i$ is defined by the macroscopic deformation $\overline{F}$ in the corresponding integration point using the relation $\bar{\mathbf{u}}_i = \overline{F} \cdot v_i$ while $\tilde{\mathbf{u}}_i$ is the microscopic fluctuation field, which is finally computed by solving the microscopic problem.

We apply periodic boundary conditions to the fluctuation and the rotations, that is,

$$\tilde{\mathbf{u}}^+ = \tilde{\mathbf{u}}^- \quad \text{and } \boldsymbol{\theta}^+ = \boldsymbol{\theta}^-$$

for each pair of periodic nodes v^+ and v^- at opposing faces of the RVE. To obtain a regular matrix K, in each corner of the RVE, the fluctuations are fixed with zero Dirichlet boundary conditions.

2.3 Macroscopic problem

We formulate the macroscopic problem based on the weak formulation of the momentum balance equation which is given by

$$\int_{\overline{B}_0} \delta\bar{x} \left(\mathrm{Div}_{\bar{x}} \overline{P}(\overline{F}) - \bar{f}\right) d\bar{x} = 0$$

for a test function $\delta\bar{x}$. Without the consideration of any volume force the equation is reduced to

$$\int_{\overline{B}_0} \delta\bar{x}\, \mathrm{Div}_{\bar{x}} \overline{P}(\overline{F}) d\bar{x} = 0. \tag{1}$$

The principle behind the FE^2 method is that the relation between the deformation gradient $\overline{F}(\bar{x})$ and the macroscopic Piola–Kirchhoff stress $\overline{P}(\overline{F}(\bar{x}))$ in a certain point $\bar{x}$ is not modeled by a material law but by the results of microscopic simulations incorporating the micro-heterogeneous structure of the respective material. While the macroscopic deformation gradient $\overline{F}(\bar{x})$ defines the boundary conditions of an attached RVE as described in the latter section, the volumetric average over the microscopic Piola–Kirchhoff stresses in the RVE yield the macroscopic stress $\overline{P}(\overline{F}(\bar{x}))$. Therefore, in each macroscopic integration point $\bar{x}$, we obtain

$$\overline{P}(\overline{F}(\bar{x})) = \frac{1}{V}\int_{B_0} P(F)dV \tag{2}$$

where $V = |B_0|$ is the volume of the RVE belonging to $\bar{x}$. We now have to specify what the integral actually means in the case of an RVE modeled with beams. In that case, the integral over the RVE is computed as the sum over all beam elements. For each beam element, the integration of the first Piola–Kirchhoff stress is based on a procedure given in [64]; see below for further details. We thus obtain

$$\overline{P}(\overline{F}(\bar{x})) = \frac{1}{V}\sum_{e=1}^{n_B}\int_{V_e} P(F)dV.$$

Let us now describe the procedure of averaging the stresses within a single beam as given in [64]. We first consider a single beam element $e = [v_i, v_j]$ which is aligned with the unit vector $r = \left(r_x, r_y, r_z\right)^T = \frac{v_j - v_i}{L_e}$. Here, L_e is the length of the considered beam element e. It is well-known that the first Piola–Kirchhoff stress can be expressed in terms of $P = J\,\sigma\,F^{-T}$, where σ is the Cauchy stress tensor and J is the determinant of the deformation gradient F. With the formulation of the locally transformed deformation $\tilde{u}$ as a cubic polynomial with respect to the normalized distance to the ends of the beam ξ, it is possible to derive the expressions $J = 1 + \operatorname{tr}(\nabla u)$ and $F^{-1} = \frac{1}{J}\left((1+J)I - F\right)$. Appendix A of the appendix presents a detailed derivation of these relations. Based on these equations we derive a formulation for the integral of the Piola–Kirchhoff stress over the volume of the beam element. This integral can be expressed as the sum of three separate integrals as

$$\begin{aligned}\int_{V_e} P dV &= \int_{V_e} \sigma\left((1+J)\,I - F^T\right)dV \\ &= \int_{V_e}\sigma dV + \int_{V_e} J\sigma dV - \int_{V_e}\sigma F^T dV.\end{aligned} \tag{3}$$

The computation of the first integral of the right-hand side is carried out as in [64]. There, a beam network is considered and a virtual work approach for the computation of an average stress for the network is introduced. The resulting expression for the average stress considers only the forces that appear in each beam element but not the higher moments. With this simplification the approach is not fully in alignment with the Euler–Bernoulli beam theory. However, it is to expect that the macroscopic behavior would not change significantly by the use of a more complex approach for the computation of the average stress. By applying this virtual work approach the first integral can be reformulated as

$$\begin{aligned}\int_{V_e}\sigma dV &= \int_{V_e}\begin{pmatrix}\sigma_{xx} & \sigma_{xy} & \sigma_{xz}\\ \sigma_{xy} & \sigma_{yy} & \sigma_{yx}\\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz}\end{pmatrix}dV \\ &= \int_{V_e} L_e\begin{pmatrix} Q_x r_x & \frac{1}{2}(Q_x r_y + Q_y r_x) & \frac{1}{2}(Q_x r_z + Q_y r_z)\\ \frac{1}{2}(Q_x r_y + Q_y r_x) & Q_y r_y & \frac{1}{2}(Q_y r_z + Q_z r_y)\\ \frac{1}{2}(Q_x r_z + Q_z r_x) & \frac{1}{2}(Q_y r_z + Q_z r_y) & Q_z r_z\end{pmatrix}dV\end{aligned} \tag{4}$$

with the forces Q_x, Q_y, and Q_z being expressed depending on the derivative of the projected deformation as $Q_{x_k} = -\frac{E\cdot I}{L_e^3}\cdot\frac{d^3\tilde{u}_{x_k}}{d\xi^3}$.

Before deriving expressions for the other two integrals of the right-hand side of Eq. (3), we make the additional assumption that the deformation u is constant over all cross sections of each beam element. This does also not align with the basic Euler–Bernoulli beam theory. However, this approximation is reasonable in our case since we are only considering relatively small cross sections compared to the length of each beam and additionally, we consider small deformations and therefore only a small bending occurs in the beams. The latter two assumptions do not need to be added, as they are already a requirement for the Euler–Bernoulli beam theory. Together with the equation derived for the integrated Cauchy stress we are able to formulate a simple expression for each of the remaining integrals. We finally obtain

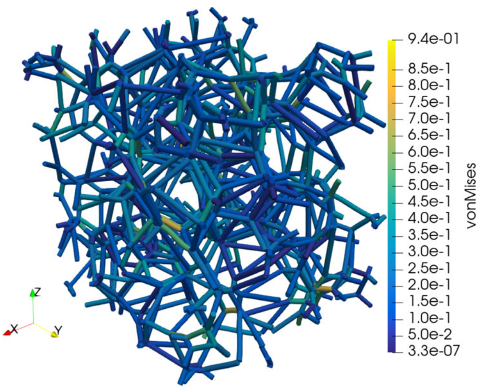
$$\begin{aligned}\int_{V_e} J\sigma \mathrm{d}V &= L_e\int_0^1 J\,\mathrm{d}\xi\int_{A_e}\sigma\,\mathrm{d}A \\ &= \left(1 + \frac{\langle u(v_2) - u(v_1),\, v_2 - v_1\rangle}{L_e}\right)\int_{V_e}\sigma\,\mathrm{d}V \text{ and}\end{aligned} \tag{5}$$

$$\begin{aligned}\int_{V_e}\sigma F^T\,\mathrm{d}V &= \int_{V_e}\sigma(I + (\nabla u)^T)\,\mathrm{d}V \\ &= \int_{V_e}\sigma\,\mathrm{d}V\cdot\left(I + \frac{(u(v_2) - u(v_1))\otimes(v_2 - v_1)}{L_e}\right)\end{aligned} \tag{6}$$

with $\otimes$ being the dyadic product.

Due to the assumption that the deformation u is constant over cross sections, it is possible to separate the volume integral into one integral over the surface of the cross section regarding the Cauchy stress σ and one integral over the length of the beam regarding the variables J and F, which both depend on the deformation u. Summing Eqs. (4), (5), and (6), finally yields an expression for the complete integral on the left-hand side of Eq. (3). One example for the computation of the Piola–Kirchhoff stress tensor for one microscopic deformation is presented in Fig. 4.

After computing the average stress in each beam, the average stress within the RVE $\overline{P}$ can be computed using Eq. (2) by summing over all beams. An algorithmic representation of our homogenization approach is shown in Fig. 5.



$$\overline{F} = \begin{pmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\downarrow$$

$$\overline{P} = \begin{pmatrix} -123.82 & 0.59 & 0.53 \\ 0.66 & -6.28 & 1.91 \\ 0.59 & 1.90 & -9.92 \end{pmatrix}$$

Fig. 4 Result of three-dimensional beam frame model that is compressed in x direction. Image of the deformed RVE on the left and the deformation gradient with the resulting averaged first Piola–Kirchhoff stress tensor on the right. Colors of the beams represent the von Mises stress values in the corresponding elements. von Mises stress values are given in megapascal (MPa)

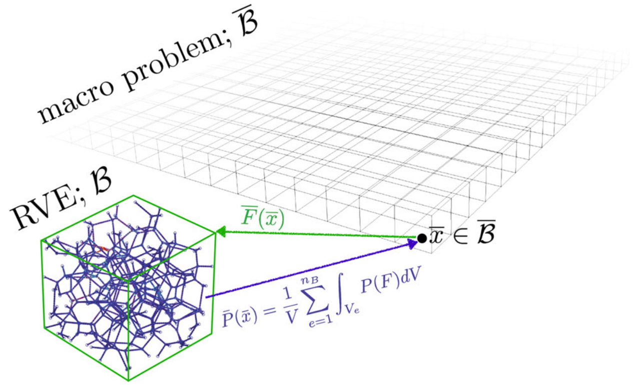

macro problem; $\overline{\mathcal{B}}$
RVE; $\mathcal{B}$
$\overline{F}(\overline{x})$
$\overline{x} \in \overline{\mathcal{B}}$
$\overline{P}(\overline{x}) = \frac{1}{V}\sum_{e=1}^{n_B}\int_{V_e} P(F)dV$

Fig. 5 FE^2 method with an RVE modeled with beams. The macroscopic deformation gradient $\overline{F}(\bar{x})$ defines the boundary conditions of the beam frame model attached to the macroscopic integration point $\bar{x}$. The stresses in the beams after solving the RVE problem is averaged and results in the macroscopic stress $\overline{P}(\overline{F}(\bar{x}))$

2.4 Algorithmic description

The macroscopic problem given in Eq. (1), which is discretized using finite elements, is solved using BFGS or an alternative (Quasi)-Newton method where the Jacobian is approximated using central difference quotients. For stability and robustness, a dynamic load stepping is integrated such that the total macroscopic deformation can be applied incrementally in several pseudo-time or load steps, respectively. When integrating over the macroscopic finite elements using a Gauss quadrature rule, for each integration point the microscopic problem is solved to obtain the corresponding value of $\overline{P}$. Let us recapitulate that we use a linear beam frame model on the microscale and thus also the resulting macroscopic problem is actually linear. Using the exact Newton method as a macroscopic nonlinear solver would therefore lead to its convergence in a single step. However, since we have put no efforts in finding an exact formulation for a consistent tangent modulus $\frac{\partial \overline{P}(\overline{F}(\bar{x}))}{\partial \overline{F}(\bar{x})}$ we cannot use an exact Newton method. Instead, as already mentioned, we use the Quasi-Newton method BFGS and can avoid computing $\frac{\partial \overline{P}(\overline{F}(\bar{x}))}{\partial \overline{F}(\bar{x})}$ completely. As an alternative, we can approximate the Jacobian matrix using difference quotients, which is an alternative Quasi-Newton approach. For the approximation we use central differences of the form

$$\left(\frac{\partial \overline{P}(\overline{F}(\bar{x}))}{\partial \overline{F}(\bar{x})}\right)_{i,j} \approx \frac{\overline{P}(\overline{F}(\bar{x}) + \epsilon_j)_i - \overline{P}(\overline{F}(\bar{x}) - \epsilon_j)_i}{2 \cdot \epsilon}$$

for $i, j \in 1, \ldots, 4$ or $i, j \in 1, \ldots, 9$. For our applications the approximation has shown to yield robust results for $\epsilon = 1e-6$. Let us remark that for the surrogate model introduced later on, we easily can use the exact Newton method and alternatively BFGS. Let us finally give a brief overview of the algorithm in Fig. 6, where we describe the procedure within one load step of the dynamic load stepping procedure.

3 Neural network-based surrogate model

Depending on the number of beam elements in the RVE, solving the microscopic problem and evaluating Eq. (2) can be computationally expensive. This becomes especially important for three-dimensional problems since a three-dimensional RVE naturally needs to consist of a much larger number of beam elements. Additionally six degrees of freedom instead of three are necessary, which also leads to larger systems of equations. To deal with this issue, we introduce a surrogate model that is supposed to compute the average Piola–Kirchhoff stress and replace the microscopic simulations based on the beam frame model. Here, we aim for training a machine learning-based surrogate model. More precisely, we have developed an artificial neural network (NN) which is trained to predict the average Piola–Kirchhoff stress tensor for one fixed microscopic structure. The NN has to be evaluated in each integration point of the macroscopic finite element problem and the input of the NN is always the deformation gradient $\overline{F}(\bar{x})$. The open-source package TensorFlow [65] is used for the development and the training of each regarded NN in this section.

Fig. 6 FE^2 algorithm using BFGS or an alternative Quasi-Newton approach as a solver on the macroscopic scale and beam frame RVEs. The complete algorithm is usually embedded in a (dynamic) load stepping scheme for robustness and global convergence

```
Init macroscopic deformation $\bar{u}^{(0)}$ using current load
/*current load is defined by dynamic load stepping*/
Loop over $k$ until convergence
  Loop over all integration points $\bar{x}$
      Compute $\overline{F}(\bar{x})$ from $\bar{u}^{(k)}$
      Apply periodic boundary conditions to RVE defined by $\overline{F}(\bar{x})$
      Solve RVE
      /*solving a beam frame problem*/
      Compute $\overline{P}$ from RVE solution
  EndLoop
  Assemble macroscopic residual vector using $\overline{P}$
  Compute Quasi-Newton (e.g. BFGS) update $\delta\bar{u}^{(k)}$
  Update $\bar{u}^{(k+1)} = \bar{u}^{(k)} + \delta\bar{u}^{(k)}$
  Check for convergence of Quasi-Newton method
EndLoop
```

3.1 Surrogate model in two dimensions

Let us first describe the training procedure and give some details on the NN architecture we use. In a first step, the fixed microscopic beam frame problem is generated with the approach explained in Sect. 2. We consider a two-dimensional open-porous structure and randomly set pores following a given pore size distribution. The resulting beam frame structure which is used to generate the training data for the model consists of 225 beam elements and 172 joints at which the beams are connected. Let us note that the model is comparably small and in the present article we just aim for a proof of concept of the suggested method. Also, we only use a prototype MATLAB implementation. It is planned to consider larger and more representative RVEs in the future using more efficient C/C++ based implementations.

For the two-dimensional RVE the data set for the training of the NN consists of about 110,000 pairs of macroscopic deformation gradients $\overline{F}$ and stresses $\overline{P}$, where the latter one is obtained by solving the corresponding RVE problem and averaging over the stresses within the beams. To obtain a large variety of different macroscopic deformation gradients, several FE^2 simulations following the algorithm from Fig. 6 have been carried out and for each BFGS step in all integration points the pairs $(\overline{F}, \overline{P})$ are stored.

For setting up the different FE^2 simulations we defined seven different macroscopic basic test cases and further vary these by changing the degree of deformation for each case. Each of the seven basic test cases has different macroscopic boundary conditions and a different deformation of the material such that the data set is expected to cover a wide variety of microscopic deformations. The magnitude and direction of each computed macroscopic deformation is set randomly during the generation process. Results for the deformation examples that are used for the generation are presented in Fig. 7. The same relatively small grid is used for each of the examples and the resulting problem has 50 degrees of freedom. There is no need to use larger problems for the generation of the training data. In contrast, the computation of a larger number of small problems yields more diversity among the localized deformation gradients in the integration points compared with a smaller number of large problems. Therefore, we expect a higher variety in the generated training data defining and using many small macroscopic problems.

In the generation of the training data we have exclusively used linear finite elements to discretize the macroscopic problem. However, the choices of the basis functions and elements for the macroscopic finite element method are not expected to affect the quality of the training data since the generated input and output values of the NN are only related to the microscopic problem. The validation data set consisting of about 12,000 input and output pairs has been generated in the same manner.

The distribution of the generated data set is presented in Fig. 8. All input components are distributed relatively tight around zero. The corresponding small values for the standard deviations which can be observed in Table 1 are intended since we are only assuming small deformations. This assumption is necessary because the applied beam frame model which is based on Euler–Bernoulli beam theory is only useful for small deformations. This is acceptable because a large proportion of reported literature investigates the linear elastic properties of open-porous materials owing to their dependence on the density that is demonstrated in terms of scaling laws, particularly between Young's modulus or compressive strength versus density [1, 66].

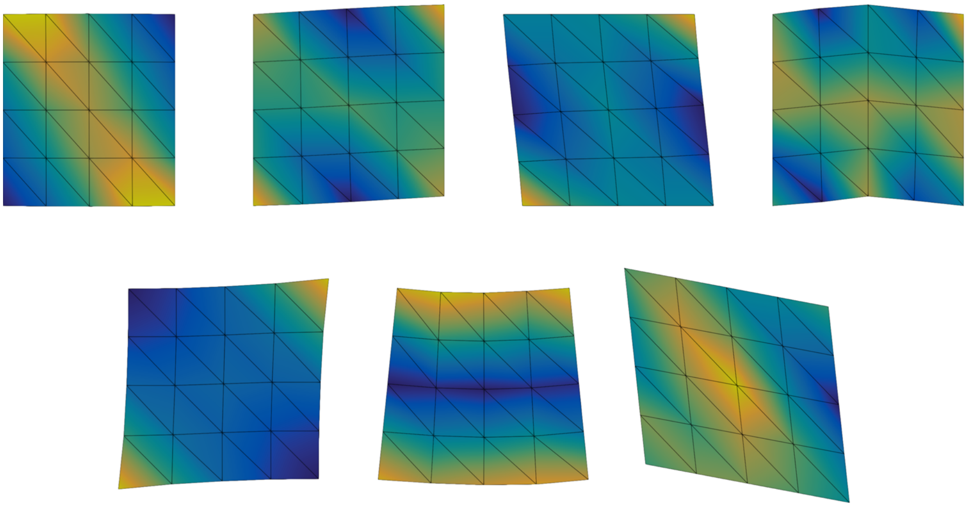


Fig. 7 Seven basic deformations that are used to generate the training data in two dimensions. To obtain a large training data set, the basic deformations are modified randomly by changing the grade and orientation of deformation. The colors represent the von Mises stresses in the computed solutions

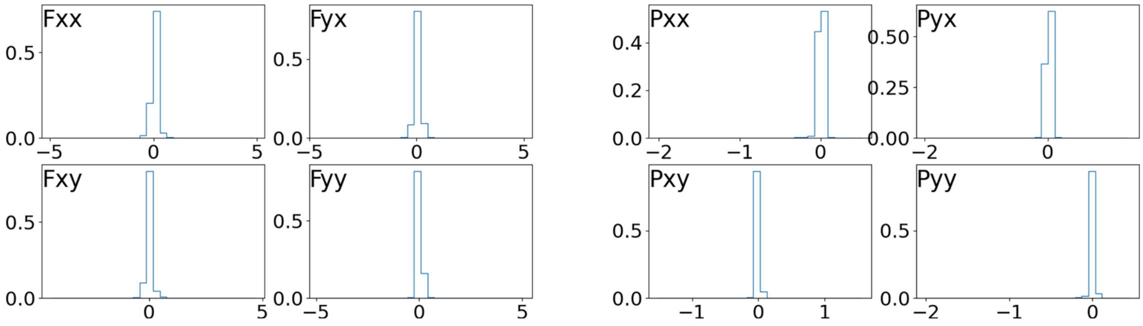


Fig. 8 Distribution of the input and output data generated for the training of the two-dimensional neural network

Table 1 Description of the input data for the training of the two-dimensional neural network

Variable name	$\overline{F}_{xx}$	$\overline{F}_{yx}$	$\overline{F}_{xy}$	$\overline{F}_{yy}$
Mean	0.02	0.03	0.00	0.00
Standard deviation	0.24	0.23	0.23	0.19
Minimum value	−4.90	−4.54	−4.31	−4.84
Maximum value	4.87	4.92	4.65	4.99

Table 2 Description of the output data for the training of the two-dimensional neural network

Variable name	$\overline{P}_{xx}$	$\overline{P}_{yx}$	$\overline{P}_{xy}$	$\overline{P}_{yy}$
Mean	0	0	0	0
Standard deviation	0.09	0.04	0.04	0.07
Minimum value	−2.00	−1.97	−1.50	−2.00
Maximum value	0.50	1.31	1.55	0.44

Table 3 Correlation coefficients between the input and output variables for the two-dimensional data set

	$\overline{P}_{xx}$	$\overline{P}_{yx}$	$\overline{P}_{xy}$	$\overline{P}_{yy}$
$\overline{F}_{xx}$	**0.39**	0.09	0.20	0.10
$\overline{F}_{yx}$	0.07	**0.50**	**0.46**	0.05
$\overline{F}_{xy}$	0.07	**0.35**	**0.52**	0.16
$\overline{F}_{yy}$	0.00	0.18	0.22	**0.28**

The most significant values are marked in bold face

As illustrated in Table 2, the components of the resulting first Piola–Kirchhoff stress tensors which are the target output variables for the training of our neural network are similarly distributed around the mean of zero. This is not surprising since the matrix in Table 3 presents distinct correlations between the input and output components.

For the two-dimensional RVE a feed-forward NN with three hidden layers is used as the basic structure of our model (see Fig. 9). The activation function for each hidden layer is chosen to be the Gaussian error linear unit (GELU) activation function [67] and the number of neurons in the hidden layers are set to 128, 256, and 128. The activation of the output is linear. For choosing a proper architecture for the NN multiple activation functions in combination with different layer sizes were considered. A grid search evaluation covering these parameters has shown that GELU activation suits especially well for the given problem and that a higher number of neurons in the model can lead to a smaller validation error. However, the reduction of the error is only significant up to 256 neurons per layer. The grid search results are

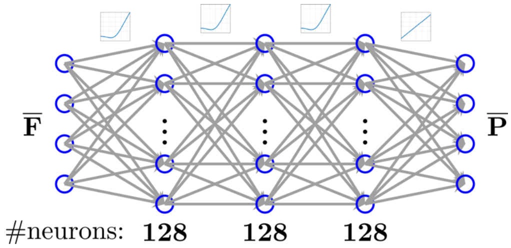


Fig. 9 Architecture of the feed-forward neural network for a two-dimensional microscopic problem. First three layers have GELU activation and linear activation function is used for the output layer

Table 4 Grid search results (test loss) for the neural network in two dimensions with different numbers of layers and neurons per layer

Activation	Neur./layer	One layer	Two layers	Three layers
Sigmoid	64	2.24e−04	8.99e−06	5.06e−06
	128	3.36e−04	3.37e−06	9.68e−07
	256	7.95e−04	1.18e−06	6.46e−07
	512	8.13e−04	4.52e−07	6.23e−07
tanh	64	3.20e−05	1.66e−06	6.68e−07
	128	3.36e−05	2.65e−06	1.03e−06
	256	9.12e−05	2.42e−06	4.20e−06
	512	1.26e−04	1.41e−06	3.62e−06
gelu	64	2.19e−07	1.69e−08	7.13e−09
	128	1.27e−07	2.58e−08	**5.76e−09**
	256	8.11e−08	8.53e−08	8.36e−09
	512	5.33e−08	1.85e−07	1.08e−08

The lowest loss value is marked in bold face

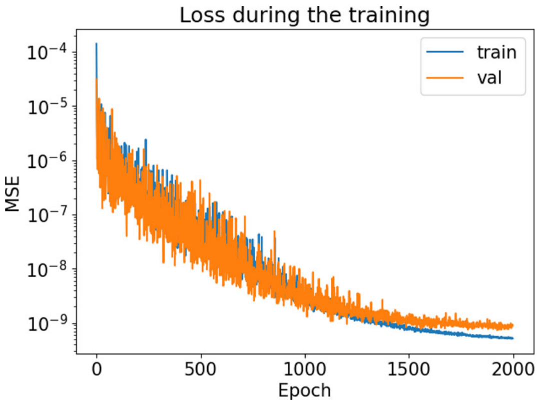


Fig. 10 Loss during the training of the neural network for two-dimensional RVEs

presented in Table 4 and the smallest values in the resulting validation loss are highlighted. The table shows that a network architecture of 3 layers with 128 neurons per layer yields the lowest loss.

For training, an adam optimizer [68] has been used to minimize the mean squared error (MSE). After a total of 1500 training epochs a sufficient reduction of the loss has been reached; see Fig. 10. The loss for the validation data has also decreased sufficiently.

3.2 Surrogate model in three dimensions

For a given three-dimensional aerogel RVE the general procedure of generating training data and training the NN works similar to the two-dimensional case. The size of the input and output of the NN is different since the three-dimensional deformation gradient and Piola–Kirchhoff stress tensor have nine components each. We consider a microscopic RVE of 1482 beam elements which are connected at 1005 joints.

The training data is similarly generated from solving macroscopic deformation test cases. However, for the three-dimensional model the approach on how to set up the macroscopic tests differs since we do not use a fixed number of deformation examples. For the generation of the data set a relatively small cube geometry is considered with 192 degrees of freedom. Dirichlet boundary conditions are applied to each of the nodes on the boundaries of this geometry with a fixed deformation determined by a randomly generated deformation gradient. This means that the deformation on the boundaries is determined by $\bar{u} = \overline{F}_d \cdot \bar{x}$ with the matrix $\overline{F}_d$ being randomly generated. Due to the large influence of randomness, the procedure is expected to yield a high variety of deformations. For the training data about 118,000 data points are generated. The validation data consists of about 13,000 equally generated data points.

The input variables are distributed around zero similar to the two-dimensional data set. The distribution of the input and output data can be visually observed in Fig. 11 and the values are also presented in Section B of the appendix. The correlation between the input and output data is presented in Table 5.

Besides the number of neurons in the input and output layer, the network architecture for the three-dimensional case is quite similar to the two-dimensional case. Here, the NN has three hidden layers with 256 neurons in each layer and again uses a GELU activation for each of these hidden layers. The choice of the architecture resulted again from a grid search evaluation. In Table 6 the resulting validation losses of the grid search algorithm is presented with the smallest values being highlighted. The grid search evaluation shows similar results to the two-dimensional case with the GELU activation function yielding the smallest loss values.

During the training of the selected neural network architecture the training loss reduced within 2000 epochs to a final loss of 4.8e−10. The development of the loss during training is presented in Fig. 12. As the image also shows, the validation loss has also decreased during the training though it settles on a slightly higher level compared to the training loss.

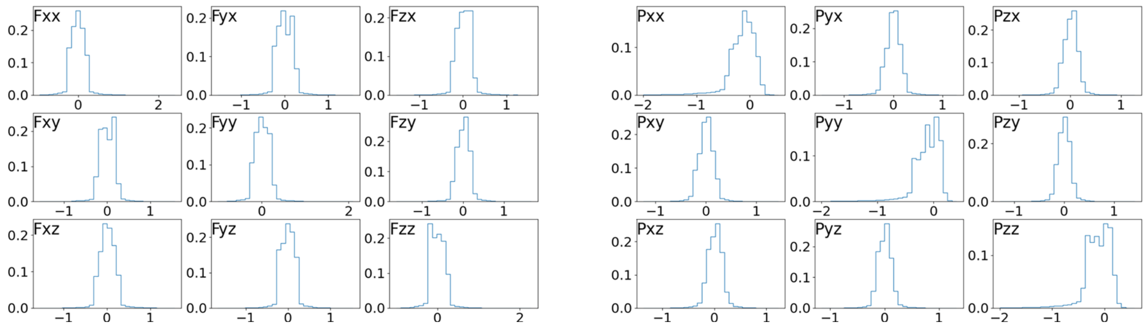


Fig. 11 Distribution of the input and output data generated for the training of the three-dimensional neural network

Table 5 Correlation coefficients between the input and output variables for the three-dimensional data set. The most significant values are marked in bold face

	$\overline{P}_{xx}$	$\overline{P}_{yx}$	$\overline{P}_{zx}$	$\overline{P}_{xy}$	$\overline{P}_{yy}$	$\overline{P}_{zy}$	$\overline{P}_{xz}$	$\overline{P}_{yz}$	$\overline{P}_{zz}$
$\overline{F}_{xx}$	**0.59**	−0.01	0.07	−0.00	−0.11	0.02	0.06	0.02	−0.02
$\overline{F}_{yx}$	−0.00	**0.68**	0.05	**0.66**	−0.02	−0.08	0.05	−0.08	0.00
$\overline{F}_{zx}$	0.06	0.07	**0.63**	0.07	0.05	−0.05	**0.61**	−0.05	0.03
$\overline{F}_{xy}$	−0.01	**0.61**	0.03	**0.63**	0.05	−0.10	0.03	−0.10	−0.03
$\overline{F}_{yy}$	−0.08	0.06	0.01	0.07	**0.59**	−0.04	0.00	−0.05	−0.04
$\overline{F}_{zy}$	−0.00	−0.04	−0.03	−0.04	−0.03	**0.62**	−0.03	**0.60**	−0.07
$\overline{F}_{xz}$	0.01	0.04	**0.61**	0.04	−0.05	0.03	**0.64**	0.03	0.09
$\overline{F}_{yz}$	−0.01	−0.06	0.01	−0.07	−0.07	**0.63**	−0.00	**0.66**	0.06
$\overline{F}_{zz}$	−0.02	0.01	0.10	0.01	−0.03	0.02	0.10	0.01	**0.64**

Table 6 Grid search results (training loss) for the neural network in three dimensions with different numbers of layers and neurons per layer

Activation	Neur./layer	One layer	Two layers	Three layers
Sigmoid	64	1.99−04	3.31e−05	2.49e−05
	128	4.55e−04	9.18e−06	6.43e−06
	256	5.50e−04	4.37e−06	2.93e−06
	512	1.37e−03	2.00e−06	2.61e−06
tanh	64	5.22e−05	1.88e−05	8.03e−06
	128	5.81e−05	1.52e−05	3.64e−06
	256	1.47e−04	1.56e−05	4.21e−06
	512	4.17e−04	1.16e−05	7.02e−06
gelu	64	1.61e−06	8.66e−07	5.07e−07
	128	3.10e−07	1.24e−07	1.34e−07
	256	1.88e−07	5.38e−08	**3.95e−08**
	512	3.38e−07	3.30e−07	4.24e−08

The lowest loss value is marked in bold face

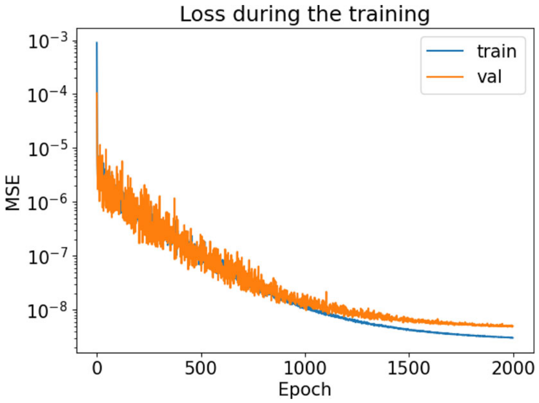


Fig. 12 Loss during the training of neural network for three-dimensional RVEs

3.3 Reduction of computational effort

The surrogate model has been originally built to reduce the computational effort for computing the first Piola–Kirchhoff stress. First of all it needs to be mentioned that the presented computation times in this work have been measured with MATLAB and that a more efficient implementation with a better use of the hardware could yield significantly different computation times. We would like to compare the computational effort of computing the first Piola–Kirchhoff stress tensor from the the NN with the effort of solving the linear system of equations presented in Sect. 2. We do not take the computational cost of training the NN into account since the training process needs to be executed only once for each machine learning model and can then be applied to run many simulations. Therefore, the training cost of the model becomes negligible compared to the reduced computational effort if the regular beam frame solver on the microscopic scale is replaced by the NN approach. The training of each NN took about 1.5 h on one 2.1 GHz core of an Intel Xeon Gold 6230 node.

In the special linear case of the linear beam frame model it is possible to apply different methods for solving the linear system of equations on the microscopic scale. Since the stiffness matrix resulting from the beam frame is the same for each microscopic problem it is possible to compute a factorization (with pivoting) of this matrix $P \cdot K = L \cdot U$ with L being a lower triangular matrix, U being a upper triangular matrix, and P being a pivoting matrix. Now, instead of solving the linear system $K \cdot u = F$ directly, the solution u can also be computed using a forward and backward substitution with the matrices L and U which yields a significantly reduced computation time. However, this procedure is only suitable in the case of a linear microscopic problem. Since the NN approach is also expected to be applicable to nonlinear problems, we are also interested in a comparison of the NN with the method of solving the microscopic problem directly with MATLAB's backslash operator, which will be the only possible option for nonlinear problems. Therefore, we want to consider for the comparison of the computation times both the forward backward substitution and the solution using the backslash operator.

Directly solving the beam frame model with the backslash operator and computing the homogenized Piola–Kirchhoff stress tensor for the two-dimensional RVE takes about 400 times longer than the evaluation of the NN with the backslash operator requiring the majority of the computation time. Since the backslash operator, that is, using the efficient UMFPACK factorization package, is generally regarded as a fast option for solving a linear system of equations, it is expected that with another implementation the computational effort of the NN solver would still be similarly smaller than the effort of the beam frame solver for this specific example. Computing the microscopic solution with a forward backward substitution is much faster than the backslash operator but also takes about 10 times longer than the evaluation of the NN.

For the three-dimensional case the factor is much higher since the beam frame RVE consists of more beam elements and solving the resulting system requires a higher computational effort. In this case, the evaluation of the NN takes about 1000 times less time than solving of the beam frame model directly with the backslash operator. This includes also the time needed in MATLAB for averaging the resulting stresses however the backslash requires the majority of the computational effort. Let us remark that in realistic applications larger RVEs will be needed. Since the computing time of sparse direct solvers as UMFPACK grows approximately cubically with the dimension of the problem, we expect an even larger benefit using NN-based surrogate models. Computing the solution with forward backward substitution also requires about 80 times the computation time of the evaluation of the NN.

```
Init macroscopic deformation $\bar{u}^{(0)}$ using current load
/*current load is defined by dynamic load stepping*/
Loop over $k$ until convergence
  Loop over all integration points $\bar{x}$
      Compute $\overline{F}(\bar{x})$ using $\bar{u}^{(k)}$
      Evaluate NN with input $\overline{F}(\bar{x})$ to obtain output $\overline{P}$
  EndLoop
  Assemble macroscopic residual vector using $\overline{P}(\overline{F}(\bar{x}))$
  Compute BFGS or Newton update $\delta\bar{u}^{(k)}$
  Update $\bar{u}^{(k+1)} = \bar{u}^{(k)} + \delta\bar{u}^{(k)}$
  Check for convergence of BFGS or Newton
EndLoop
```

Fig. 13 FE^2 algorithm using the Newton or the BFGS method as a solver on the macroscopic scale and an NN as a surrogate model on the microscopic scale. The complete algorithm is usually embedded in a (dynamic) load stepping scheme for robustness and global convergence

3.4 Algorithmic description

Basically, the resulting algorithm exploiting the surrogate model is similar to Algorithm 6. Only the evaluation of the average stress $\overline{P}$ is replaced by an evaluation of the NN. Let us remark that by deriving the NN, we can easily compute the tangent/Jacobian matrix and alternatively use Newton's method instead of BFGS here. We summarize the algorithm in Fig. 13.

4 Numerical results

In this section, we aim for a numerical verification that the developed homogenization approach using the beam frame model gives reasonable results and that the machine learning-based surrogate model accurately approximates them in a computationally efficient manner. Additionally, we investigate the convergence behavior of the BFGS and Newton method for different examples. All stress values presented in this section are given in megapascals (MPa).

4.1 Microscopic simulation and NN prediction

First, we want to illustrate the behavior of the NN in comparison to the beam frame solver with the homogenization of the Piola–Kirchhoff stresses by presenting some results for microscopic deformations, the resulting Piola–Kirchhoff stress computed with the homogenization method, and the prediction of the NN for the same deformation. We consider the three-dimensional RVE which is also shown in figure 4 with the microscopic boundary conditions that are introduced in Sect. 2. The results of uniaxial compression tests in each of the three dimensions and two examples which show a shift of the RVE. For the presented microscopic deformations the predictions of the NN are very close to the Piola–Kirchhoff stress tensors computed with the beam frame solver and the homogenization method. Table 7 also shows the norm of the deviation between both computed stress tensors. The Frobenius norm is used for the evaluation.

4.2 Convergence of the macroscopic solver

As already mentioned we use (Quasi)-Newton methods [69] for solving the macroscopic problem. Exact Newton methods require the computation of a tangent which is unknown for the case of the beam RVEs and its computation is expected to be expensive. The BFGS method is a common choice to overcome the problem of computing the tangent since it requires only an initial guess for the tangent and adjusts this approximation in each iteration using only first order derivatives, as, e.g., the gradient. In general, this approach lacks the quadratic order of convergence of Newton's method [70], but each iteration can be computed much faster. In the present work, we consider Newton's method with an exact tangent for the NN approach and a Quasi-Newton method with an approximate tangent for the beam frame approach. The numerical approximation in this case is computed with central differences. In the following, we will refer to this approach as Newton's method with beam frame microstructure. We also consider the BFGS method [69, 71] for both NN and beam frame approach with the Wolfe [72] condition to control the step length. We initialize the Hessian of the first BFGS step with the exact Hessian in the case of the NN and the approximated Hessian in the beam case. This increases the computation time for the first BFGS step but reduces the total number of iterations for the solver. The stopping criteria for both, the Newton and BFGS are both based on the reduction of the relative residual. If the ratio between the residual of the current update and the residual of the initial value is smaller than 1e−10 the criterium is satisfied and the solver terminates.

For our first two-dimensional test case we consider a square domain on the macroscopic scale. Dirichlet boundary conditions are applied to the left and right boundaries of the domain and these nodes are shifted in x direction depending on their y value; see Fig. 14 for the macroscopic deformation. The macroscopic problem is discretized using bilinear brick finite elements or linear or quadratic triangular ones. We will use this test case to compare different aspects of our macroscopic solvers. Refining the mesh is straight forward, namely by doubling the numbers of finite elements in x and y direction which leads to a quadratic increase in the degrees of freedom (dofs). To find a solution, the solver has to adjust the position of all considered nodes as can be inferred from the solutions in Fig. 14.

Considering the beam frame model for the microscopic problem, both macroscopic solvers are capable for each considered test case to find a solution that reduces the residual until it reaches the stopping criteria. For larger test cases, an adaptive load stepping has proved to increase the stability of both methods.

In Table 8 the number of iterations per load step and the total computation time are presented for the beam frame (BF) model as well as for the surrogate NN on the microscopic scale. For solving the beam frame model on the microscopic scale we regard two methods mentioned in Sect. 3. Due to the linearity of the microscopic problem a decomposition of the stiffness matrix is possible and the microscopic problem can be solved with a forward backward substitution. In Table 8 we present the computation times for this approach as well as for the much slower approach of solving the linear system of equations directly with MATLAB's backslash solve which would also be required for a nonlinear microscopic problem. For the macroscopic discretization P1 elements are considered. The table shows that on average the BFGS method requires only slightly more iterations per load step than the Newton method. However, in contrast to the Newton method, the BFGS method does not require recomputing the tangent and solving a linear system of equations in each step. Due to this reduced computational effort for each iteration the total computation time of the BFGS method is significantly lower. For both microscopic solvers the behavior with respect to the number of iterations is identical since the NN is trained to replicate the deformation behavior and the resulting stresses of the beam frame model. Therefore, the macroscopic solver obtains a very similar information in each quadrature point and likewise computes similar iterations. Let us remark that the computing times are much lower using the surrogate model but we do not see the same factors that we saw in the comparison of the computational effort for the microscopic solvers in the previous section. This is the case since many MATLAB-based computations, especially on the macroscopic scale, are carried out in both cases in addition to the computation of $\overline{P}$. This portion of the computing time can surely be decreased by a hardware-aware and efficient implementation. Especially the macroscopic solver can either be parallelized in case of Newton's method or transferred to a GPU in case of BFGS. Domain decomposition methods (DDM) and multigrid methods (MG) are expected to reduce the computation time of the macroscopic solvers. An efficient HPC implementation of our approaches exploiting GPUs is planned for the future.

4.3 Comparison of different macroscopic grids

The NN has only been trained on simulation results computed with quadrilateral elements and bilinear Ansatz functions on the macroscopic scale. The numerical test cases use also discretizations with triangular elements and linear (P1) or quadratic (P2) Ansatz functions. To confirm that the NN can be used for P1 and P2 discretizations despite it was not trained

Table 7 Piola–Kirchhoff stress computed for the three-dimensional RVE with beam frame homogenization method and neural network

deformation gradient	$\overline{P}_{BF}$	$\overline{P}_{NN}$	$\frac{\Vert\overline{P}_{BF}-\overline{P}_{NN}\Vert_F}{\Vert\overline{P}_{BF}\Vert_F}$
$\overline{F}=\begin{pmatrix}0.9&0&0\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}-123.82&0.59&0.53\\0.66&-6.28&1.91\\0.59&1.90&-9.92\end{pmatrix}$	$\begin{pmatrix}-123.82&0.61&0.53\\0.67&-6.33&1.90\\0.60&1.87&-9.95\end{pmatrix}$	5.90e−4
$\overline{F}=\begin{pmatrix}1&0&0\\0&0.9&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}-9.79&-1.92&-0.45\\-1.98&-121.82&0.21\\-0.50&0.19&-9.53\end{pmatrix}$	$\begin{pmatrix}-9.80&-1.89&-0.46\\-1.95&-121.85&0.21\\-0.51&0.15&-9.55\end{pmatrix}$	5.09e−4
$\overline{F}=\begin{pmatrix}1&0&0\\0&1&0\\0&0&0.9\end{pmatrix}$	$\begin{pmatrix}-10.82&0.89&-1.84\\0.89&-6.54&-0.05\\-1.85&0.00&-121.05\end{pmatrix}$	$\begin{pmatrix}-10.85&0.91&-1.86\\0.89&-6.60&-0.05\\-1.86&-0.03&-121.06\end{pmatrix}$	7.39e−4
$\overline{F}=\begin{pmatrix}1&0.2&0\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}-28.01&104.40&1.41\\104.32&-8.72&-2.75\\1.39&-2.77&-8.52\end{pmatrix}$	$\begin{pmatrix}-28.02&104.44&1.39\\104.32&-8.77&-2.73\\1.38&-2.78&-8.50\end{pmatrix}$	4.63e−4
$\overline{F}=\begin{pmatrix}1&0&0.2\\0&1&0\\0&0&1\end{pmatrix}$	$\begin{pmatrix}-35.89&0.32&107.46\\0.36&-10.51&1.47\\107.43&1.51&-8.78\end{pmatrix}$	$\begin{pmatrix}-35.88&0.38&107.44\\0.40&-10.57&1.51\\107.41&1.52&-8.78\end{pmatrix}$	7.03e−4

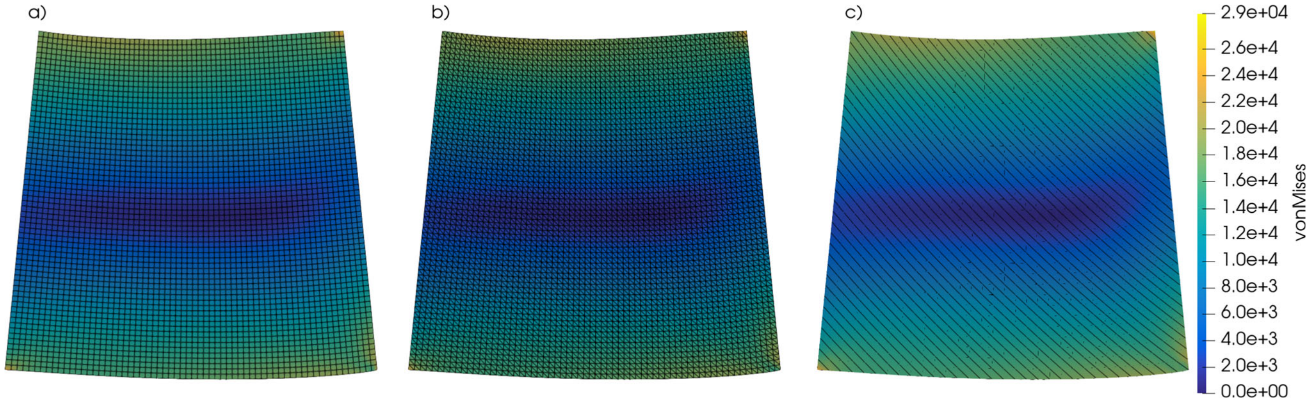


Fig. 14 Solutions of the deformation test with Q1 (**a**), P1 (**b**), and P2 (**c**) Ansatz functions

Table 8 Number of macroscopic iterations per load step and total computing time for P1 discretization

	dofs	50	162	578	2178	8450	33282
BF solver	Newton iterations	2	3	3	3	4.3	4
	BFGS iterations	4	4.5	5	7.5	12.3	12.6
	Newton comp. time [min]	0.37	2.01	4.72	15.01	138.25	709.21
	BFGS comp. time [min]	0.16	0.69	2.59	9.06	126.08	458.75
	Newton w. backslash solver [min]	1.25	10.36	45.35	247.05	2080.12	>48 h
	BFGS w. backslash solver [min]	0.39	1.86	7.01	252.94	2738.93	>48 h
NN solver	Newton iterations	2	3	3	3	4.3	4
	BFGS iterations	4	4.5	5	7.5	12.3	12.6
	Newton comp. time [min]	3e-3	0.01	0.05	0.21	4.90	402.70
	BFGS comp. time [min]	4e-3	0.01	0.04	0.18	4.78	179.71

on data obtained with P1 or P2 simulations, the deviation of the solutions computed with the beam frame model are compared to the solutions computed with the surrogate model.

The deviations of the solutions computed with the NN from the solutions computed with the beam frame model are shown in Table 9. In the tables $\bar{\mathbf{u}}_{NN}$ refers to the solution computed with the NN and $\bar{\mathbf{u}}_{BF}$ refers to the solution computed with the beam frame model. The expressions $\bar{\sigma}_{vM,NN}$ and $\bar{\sigma}_{vM,BF}$ refer to the corresponding von Mises stresses. We consider different macroscopic meshes with the level of refinement marked by the corresponding degrees of freedom (dofs).

We can observe that the errors of the NN solution computed with different Ansatz functions at the same level of refinement are very similar. The choice of the Ansatz function does not seem to have an influence on the quality of the macroscopic solution when the surrogate model is used. Let us remind that the network has only been trained on data obtained from solutions computed with bilinear square elements. Nonetheless the NN is capable to yield equally good solutions for other element shapes and Ansatz func-

Table 9 Deviations of the NN solutions and the BF solutions in different norms for different macroscopic discretizations (discr.), that is, bilinear brick (Q1) and linear (P1) and quadratic (P2) triangular elements

Norm	Discr.	Degrees of freedom					
		50	162	578	2178	8450	33282
$\frac{\|\bar{\mathbf{u}}_{NN}-\bar{\mathbf{u}}_{BF}\|_2}{\|\bar{\mathbf{u}}_{BF}\|_2}$	Q1	$1.28e-3$	$1.58e-3$	$1.73e-3$	$1.80e-3$	$1.84e-3$	$1.85e-3$
	P1	$1.24e-3$	$1.56e-3$	$1.72e-3$	$1.80e-3$	$1.83e-3$	$1.85e-3$
	P2	$1.30e-3$	$1.61e-3$	$1.74e-3$	$1.81e-3$	$1.84e-3$	$1.85e-3$
$\frac{\|\bar{\mathbf{u}}_{NN}-\bar{\mathbf{u}}_{BF}\|_\infty}{\|\bar{\mathbf{u}}_{BF}\|_\infty}$	Q1	$1.38e-3$	$1.52e-3$	$1.56e-3$	$1.58e-3$	$1.58e-3$	$1.58e-3$
	P1	$1.33e-3$	$1.49e-3$	$1.55e-3$	$1.57e-3$	$1.58e-3$	$1.58e-3$
	P2	$1.38e-3$	$1.54e-3$	$1.57e-3$	$1.58e-3$	$1.58e-3$	$1.58e-3$
$\frac{\|\bar{\sigma}_{vM,NN}-\bar{\sigma}_{vM,BF}\|_2}{\|\bar{\sigma}_{vM,BF}\|_2}$	Q1	$1.19e-3$	$1.26e-3$	$1.37e-3$	$1.43e-3$	$1.46e-3$	$1.46e-3$
	P1	$1.11e-3$	$1.28e-3$	$1.39e-3$	$1.44e-3$	$1.45e-3$	$1.46e-3$
	P2	$1.54e-3$	$1.47e-3$	$1.46e-3$	$1.47e-3$	$1.47e-3$	$1.46e-3$
$\frac{\|\bar{\sigma}_{vM,NN}-\bar{\sigma}_{vM,BF}\|_\infty}{\|\bar{\sigma}_{vM,BF}\|_\infty}$	Q1	$1.43e-3$	$1.59e-3$	$1.90e-3$	$2.39e-3$	$2.42e-3$	$3.13e-3$
	P1	$1.66e-3$	$1.80e-3$	$1.86e-3$	$2.42e-3$	$2.55e-3$	$3.13e-3$
	P2	$1.82e-3$	$1.81e-3$	$2.45e-3$	$3.16e-3$	$3.41e-3$	$2.88e-3$

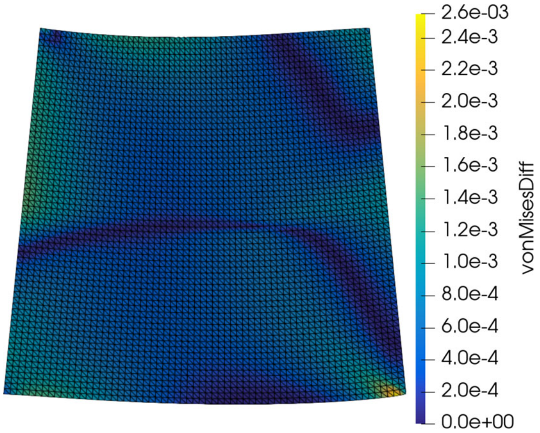


Fig. 15 Absolute difference in von Mises stress between the NN solution and the solution computed with the beam frame model

tions. In general, this is expected, since the NN is trained on localized data which does not really depend on the finite element discretization used on the macroscopic scale. Finally, the computed solutions for a refinement with 8450 dofs are presented in Fig. 14 and the absolute difference between the von Mises stress of the solutions computed with the NN and with the beam frame model is shown in Fig. 15. The macroscopic error of the NN solution is not evenly distributed and higher error values can be observed near the boundaries of the domain. However, the expected distribution of the derivation values is difficult to predict since the error depends locally on the quality of the networks prediction for the Piola–Kirchhoff stress.

The results show that the deviation of the solution computed with the NN from the solution computed with the beam frame model slightly increases with an increasing number of elements. The behavior can be explained with the fact that the computation of the Piola Kirchhoff stress in a single integration point already slightly deviates from the averaged stress computed with the beam frame model. This error is small for each evaluation but in finer macroscopic meshes with more evaluation points these small errors can sum up to a larger macroscopic deviation in the deformation. In the considered mesh refinements the error stays relatively small and for some refinements and norm measurements the error seems to stay constant or even decreases. This behavior leaves room for the assumption that the error might converge with a further refinement and might not increase significantly when a certain level is reached. Unfortunately we are not able to support this assumption with further data at this point. We plan larger simulations using a more efficient software framework than MATLAB in the future.

4.4 Compression of a punched plate geometry

As a more complex example, we consider a two-dimensional plate with a circular hole in the middle. The square shaped geometry has an edge length of 1 and the hole center has a diameter of 0.4. We want to apply a compression test on this geometry to compare the macroscopic solutions using the beam frame approach and the NN. The geometry is discretized with P1 elements for both microscopic solvers and the macroscopic solution is computed with the BFGS method. For the compression test, the left and right boundaries are shifted towards the hole and fixed in x and y direction with Dirichlet boundary conditions.

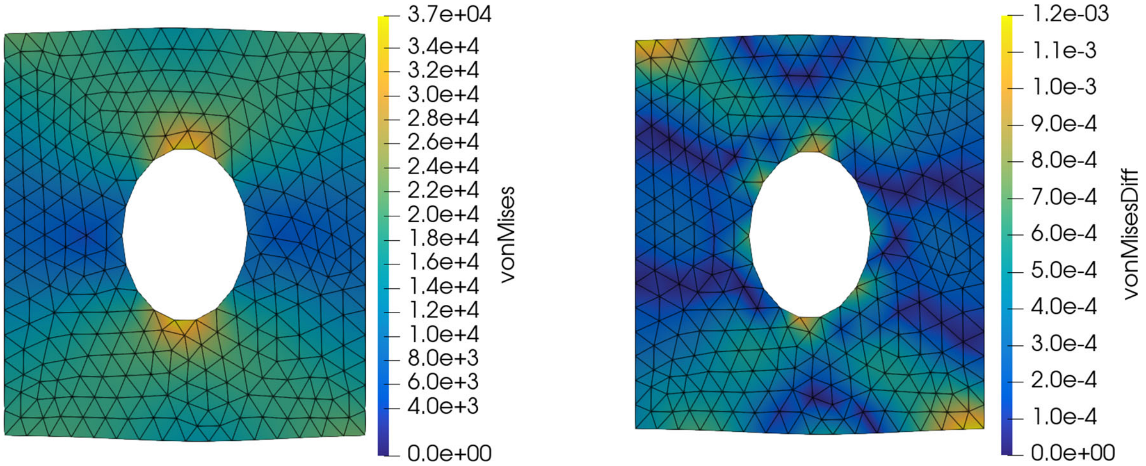


Fig. 16 Compressed punched plate computed with the NN approach and P1 elements. Coloring of the elements in the left image represents the resulting von Mises stresses and coloring in the right image shows the absolute difference of the von Mises stress to the solution computed with the beam frame solver

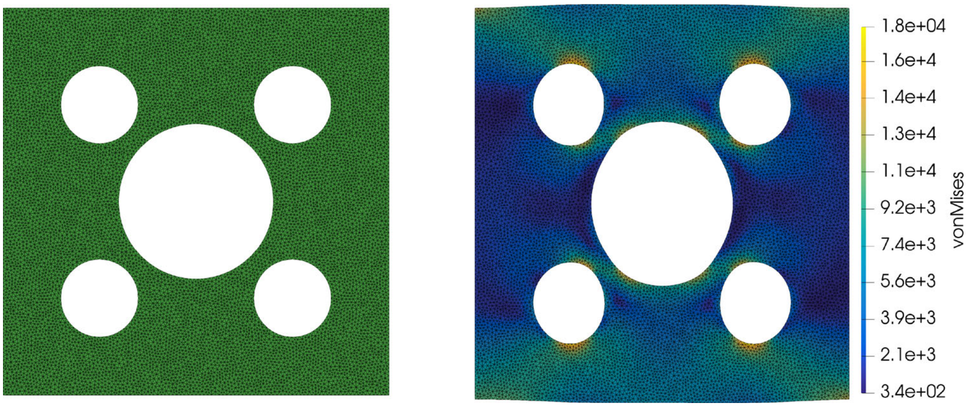


Fig. 17 Compressed plate with five punched holes computed with the NN approach and P1 elements. Colors of the elements represent the resulting von Mises stress

The result of the compression test computed with the NN approach and the resulting difference in von Mises stress are shown in Fig. 16. The largest stress differences between the two solutions are observed in the regions of the geometry with also the highest stress values. The total error between the macroscopic solutions is evaluated with the following norms:

$$\frac{\|\bar{\mathbf{u}}_{NN} - \bar{\mathbf{u}}_{BF}\|_2}{\|\bar{\mathbf{u}}_{BF}\|_2} = 1.26e - 3,$$

$$\frac{\|\bar{\mathbf{u}}_{NN} - \bar{\mathbf{u}}_{BF}\|_\infty}{\|\bar{\mathbf{u}}_{BF}\|_\infty} = 1.80e - 3,$$

$$\frac{\|\bar{\sigma}_{vM,NN} - \bar{\sigma}_{vM,BF}\|_2}{\|\bar{\sigma}_{vM,BF}\|_2} = 8.96e - 4,$$

$$\frac{\|\bar{\sigma}_{vM,NN} - \bar{\sigma}_{vM,BF}\|_\infty}{\|\bar{\sigma}_{vM,BF}\|_\infty} = 1.20e - 3.$$

4.5 Compression of a punched plate geometry with five holes

The NN approach allows the simulation of finer and more complex geometries without exhausting the computational resources. As an example we consider a two-dimensional square-shaped plate with an edge length of 1 and five circular holes. One hole is in the center of the plate with a diameter of 0.4 and four smaller holes with a diameter of 0.2 are placed around it with the same distance from the center. The circles which form the holes are placed without any overlap. For this geometry we apply the same two-dimensional compression test as in the previous example with the left and right boundaries being fixed with Dirichlet boundary conditions. We use linear (P1) elements for a fine discretization of the geometry which yields 23,824 degrees of freedom. The computed BFGS solution of this deformation is presented in Fig. 17 with the resulting von Mises stresses expressed by the coloring of the elements.

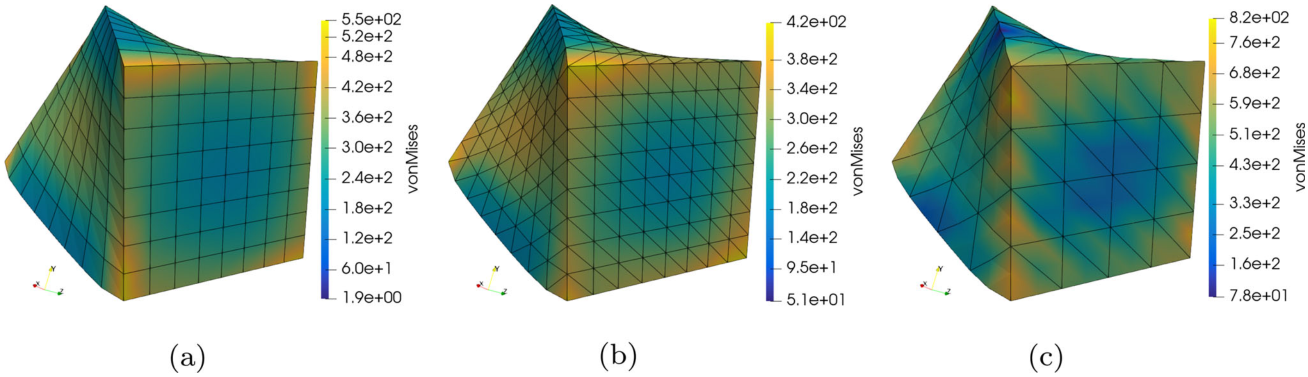


Fig. 18 Solution of the three-dimensional deformation test computed with Q1 (**a**), P1 (**b**), and P2 (**c**) Ansatz functions. Coloring of the elements represents the resulting von Mises stresses

4.6 Comparison of three-dimensional test cases

For our macroscopic test problem in three dimensions we consider a regular cube. Dirichlet boundary conditions are applied to the boundaries in x direction and the affected nodes are both rotated around the center of the face and pulled in x direction away from the center of the cube. The cube is rotated by a total of 36 degrees. This geometry again fits well for a comparison of different finite element grids since trilinear cube elements can be used for the discretization as well as linear or quadratic tetrahedral elements. The refinement of the mesh is achieved by doubling the number of elements in each dimension and yield a cubic increase in the dofs. Solutions of the test problem computed with the different Ansatz functions are presented in Fig. 18. A comparison of the error between the macroscopic solution computed with the beam frame model and the solution resulting from the surrogate approach is shown in Table 10. Multiple levels of refinement are computed and the results show that the norms of the deviations are in a very similar range to the norms of the regarded two-dimensional test case. The table shows that the errors for one refinement level is about the same for all Ansatz functions and does not indicate a better performance of the NN approach for one of the discretizations. For higher numbers of degrees of freedom the errors increase slightly, without occurrence of any significant outliers.

Table 10 Deviations of the NN solutions and the BF solutions in different norms for different macroscopic discretizations (discr.), that is, bilinear brick (Q1) and linear (P1) and quadratic (P2) tetrahedral elements

		degrees of freedom		
norm	**discr.**	108	525	3159
$\frac{\|\bar{\mathbf{u}}_{NN}-\bar{\mathbf{u}}_{BF}\|_2}{\|\bar{\mathbf{u}}_{BF}\|_2}$	Q1	$2.36e-4$	$3.56e-4$	$4.13e-4$
	P1	$1.72e-4$	$2.91e-4$	$3.77e-4$
	P2	$1.65e-4$	$3.52e-4$	$4.22e-4$
$\frac{\|\bar{\mathbf{u}}_{NN}-\bar{\mathbf{u}}_{BF}\|_\infty}{\|\bar{\mathbf{u}}_{BF}\|_\infty}$	Q1	$4.95e-4$	$6.45e-4$	$7.03e-4$
	P1	$3.84e-4$	$5.68e-4$	$6.26e-4$
	P2	$3.20e-4$	$6.83e-4$	$7.27e-4$
$\frac{\|\bar{\sigma}_{vM,NN}-\bar{\sigma}_{vM,BF}\|_2}{\|\bar{\sigma}_{vM,BF}\|_2}$	Q1	$2.98e-4$	$4.56e-4$	$5.72e-4$
	P1	$2.32e-4$	$3.59e-4$	$4.80e-4$
	P2	$2.44e-4$	$3.40e-4$	$4.83e-4$
$\frac{\|\bar{\sigma}_{vM,NN}-\bar{\sigma}_{vM,BF}\|_\infty}{\|\bar{\sigma}_{vM,BF}\|_\infty}$	Q1	$6.05e-4$	$1.11e-3$	$1.66e-3$
	P1	$4.33e-4$	$9.07e-4$	$1.56e-4$
	P2	$3.85e-4$	$6.24e-4$	$1.79e-4$

4.7 Torsion of a cube with a cylindrical hole

As a more complex example of a three-dimensional deformation, we consider a cube with an edge length of 1 and a cylindrical hole in the middle. The cylinder-shape is placed parallel to the x axis with a diameter of 0.5 and the geometry is discretized with linear P1 elements. Dirichlet boundary conditions are applied to the boundaries with x is equal to zero or one and similar to the previous test example each affected node is twisted and shifted in x direction away from the cubes center. We again consider a total rotation of 36 degrees. The result with the corresponding von Mises stresses are presented in Fig. 19.

4.8 Torsion of a cylinder

As another example of a three-dimensional deformation we consider a cylinder with the length of 2 and a diameter of 1. The cylinder-shape is placed parallel to the x axis and the geometry is discretized with linear P1 elements. The resulting discretization has 27,231 dofs. For the torsion test case Dirichlet boundary conditions are applied to the boundaries with x equals zero or one similar to the previous test example each affected node is twisted and shifted in x direction away from the cylinders center. The total rotation of the geometry is 36 degrees as in the previous test cases. The results in Fig. 20 show that the distribution of the von Mises stresses is more evenly compared to the examples of the torsion test of a cube in Fig. 18.

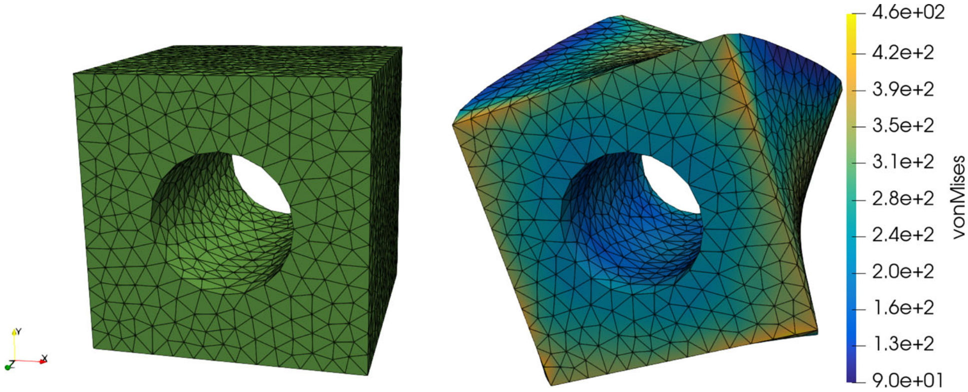


Fig. 19 Torsion of a cube geometry with a circular hole discretized with P1 elements

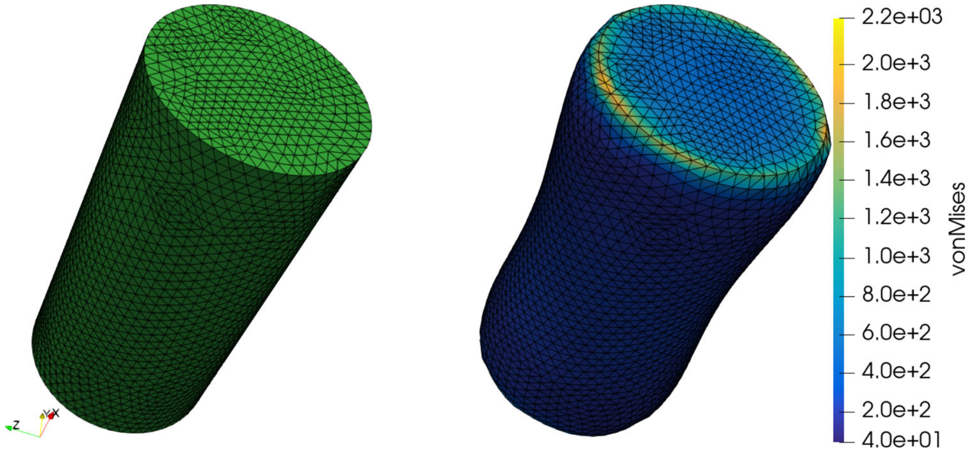


Fig. 20 Torsion of a cylinder geometry discretized with P1 elements. Left image shows the undeformed geometry and right image shows the computed BFGS solution

Appendix A Derivation of determinate and inverse for the beam frame model

Based on the assumptions of the beam frame model as described in Sect. 2 the deformation of each beam element can be expressed with respect to the distance of its starting node.

$$\mathbf{u}(x, y, z) = \tilde{\mathbf{u}}(\xi) = \begin{pmatrix} a_x\xi^3 + b_x\xi^2 + c_x\xi + d_x \\ a_y\xi^3 + b_y\xi^2 + c_y\xi + d_y \\ a_z\xi^3 + b_z\xi^2 + c_z\xi + d_z \end{pmatrix}$$

with $\xi = \frac{\|(x,y,z)^T - v_1\|}{\|v_2 - v_1\|} \in [0, 1]$ being the relative distance from the coordinate X to the first vertex of the beam v_1. With this expression it is possible to compute the Jacobian of u as

$$\nabla\mathbf{u}(X) = \begin{pmatrix} u_{xx} & u_{yx} & u_{zx} \\ u_{xy} & u_{yy} & u_{zy} \\ u_{xz} & u_{yz} & u_{zz} \end{pmatrix}$$

with $u_{x_i x_j} = \frac{\partial \tilde{u}_{x_i}}{\partial \xi}\frac{\partial \xi}{\partial x_j} = \left(3a_{x_i}\xi^2 + 2b_{x_i}\xi + c_{x_i}\right)\frac{(v_2 - v_1)_j}{\|v_2 - v_1\|^2}$. This elementwise notation is equivalent to the dyadic product $\nabla\mathbf{u}(X) = \nabla_\xi\,\tilde{\mathbf{u}}(\xi)\otimes\nabla\xi$. Therefore, the deformation gradient can be expressed in the form

$$F = I + p \otimes q.$$

for $p = \nabla_\xi\,\hat{u}(\xi)$ and $q = \nabla\xi$.

For the three-dimensional case we will now derive that for every matrix $F \in \mathbb{R}^{3\times3}$ that follows this relation and any given $p, q \in \mathbb{R}^3$ the determinant is given by $\det(F) = 1 + \operatorname{trace}(p \otimes q)$. Using Leibniz formula for the computation of the determinant of the 3×3 matrix yields

$$\begin{aligned}\det(F) =&(1 + p_x\,q_x)\cdot(1 + p_y\,q_y)\cdot(1 + p_z\,q_z)\\ &+ 2\,p_x\,q_x\,p_y\,q_y\,p_z\,q_z\\ &-(1 + p_x\,q_x)\cdot p_y\,q_y\,p_z\,q_z\\ &-(1 + p_y\,q_y)\cdot p_x\,q_x\,p_z\,q_z\\ &-(1 + p_z\,q_z)\cdot p_x\,q_x\,p_y\,q_y.\end{aligned}$$

After resolving the brackets most of the terms cancel out which leads to the simplified form

$$\det(F) = 1 + p_x\,q_x + p_y\,q_y + p_z\,q_z = 1 + \operatorname{trace}(p \otimes q).$$

With the given expression for the determinant it is also possible to derive the formulation for the inverse $F^{-1} = \frac{1}{\det(F)}\left((1 + \det(F)) - F\right)$. For the computation of the inverse

Table 11 Description of the input data for the training of the three-dimensional neural network

Variable name	$\overline{F}_{xx}$	$\overline{F}_{yx}$	$\overline{F}_{zx}$	$\overline{F}_{xy}$	$\overline{F}_{yy}$	$\overline{F}_{zy}$	$\overline{F}_{xz}$	$\overline{F}_{yz}$	$\overline{F}_{zz}$
Mean	−0.01	0.00	0.02	0.00	0.01	0.00	0.01	−0.01	0.00
Standard deviation	0.19	0.19	0.19	0.19	0.19	0.18	0.19	0.18	0.19
Minimum value	−0.96	−1.54	−1.54	−1.56	−1.01	−1.64	−1.56	−1.66	−1.01
Maximum value	2.40	1.57	1.57	1.56	2.10	1.67	1.57	1.53	2.27

Table 12 Description of the output data for the training of the three-dimensional neural network

Variable name	$\overline{P}_{xx}$	$\overline{P}_{yx}$	$\overline{P}_{zx}$	$\overline{P}_{xy}$	$\overline{P}_{yy}$	$\overline{P}_{zy}$	$\overline{P}_{xz}$	$\overline{P}_{yz}$	$\overline{P}_{zz}$
Mean	−0.16	0.00	0.01	0.00	−0.13	0.00	0.01	0.00	−0.13
Standard deviation	0.27	0.16	0.16	0.16	0.27	0.14	0.16	0.14	0.27
Minimum value	−2.00	−1.46	−1.38	−1.24	−2.00	−1.29	−1.37	−1.16	−2.00
Maximum value	0.53	1.29	1.27	1.44	0.41	1.42	1.23	1.32	0.58

we use the well-known relation of the inverse with the adjugate matrix $F^{-1} = \frac{1}{\det(F)} \cdot \operatorname{adj}(F)$ with the adjugate matrix given by

$$\operatorname{adj}(F) = \begin{pmatrix} 1 + p_y\, q_y + p_z\, q_z & -p_x\, q_y & -p_x\, q_z \\ -p_y\, q_x & 1 + p_x\, q_x + p_z\, q_z & -p_y\, q_z \\ -p_z\, q_x & -p_z\, q_y & 1 + p_x\, q_x + p_y\, q_y \end{pmatrix}.$$

Finally, together with the relation $\det(F) = 1 + p_x\, q_x \cdot p_y\, q_y \cdot p_z\, q_z$ it is possible to express the inverse matrix as

$$F^{-1} = \frac{1}{\det(F)} \operatorname{adj}(F) = \frac{1}{\det(F)} \left((1 + \det(F)) \cdot I - F \right).$$

Appendix B Distribution of the training data for the three-dimensional NN

The distribution of the training data generated for training the NN in the three-dimensional case is presented in this part of the Appendix. Table 11 shows the measures of the distribution of the input data and the distribution of the output data is described by Table 12.

Acknowledgements The authors gratefully acknowledge the Program Directorate Transport of the German Aerospace Centre for funding the Road Traffic Project FFAE. We would also like to thank Shivangi Aney (German Aerospace Centre) for providing the software to generate the open-porous RVEs and Simon Klaes (University of Cologne) for supporting the visualization of microscopic and macroscopic solutions in Paraview.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Gibson LJ, Ashby MF (1999) Cellular solids: structure and properties. Cambridge University Press. https://doi.org/10.1017/CBO9781139878326
2. Fricke J, Emmerling A (1992) Aerogels. J Am Chem Soc 75(8):2027–2036. https://doi.org/10.1111/j.1151-2916.1992.tb04461.x
3. Rege A (2023). In: Aegerter MA, Leventis N, Koebel M, Steiner SA III (eds) Modeling the structural, fractal and mechanical properties of aerogels. Springer, Cham, pp 289–305. https://doi.org/10.1007/978-3-030-27322-4_12
4. Gent A, Thomas A (1959) The deformation of foamed elastic materials. J Appl Polym Sci 1(1):107–113. https://doi.org/10.1002/app.1959.070010117
5. Gibson LJ, Ashby MF, Schajer GS, Robertson CI (1982) The mechanics of two-dimensional cellular materials. Proc R Soc Lond Ser A Math Phys Eng Scie 382:25–42. https://doi.org/10.1098/rspa.1982.0087
6. Gibson LJ, Ashby MF (1982) The mechanics of three-dimensional cellular materials. Proc R Soc Lond Ser A Math Phys Eng Scie 382(1782):43–59. https://doi.org/10.1098/rspa.1982.0088
7. Warren W, Kraynik A (1997) Linear elastic behavior of a low-density Kelvin foam with open cells. J Appl Mech 64(4): 787–794 https://doi.org/10.1115/1.2788983
8. Dement'ev A, Tarakanov O (1970) Effect of cellular structure on the mechanical properties of plastic foams. Polym Mech 6(4):519–525. https://doi.org/10.1007/BF00856299
9. Jang W-Y, Kyriakides S, Kraynik AM (2010) On the compressive strength of open-cell metal foams with Kelvin and random cell structures. Int J Solids Struct 47(21):2872–2883. https://doi.org/10.1016/j.ijsolstr.2010.06.014
10. Gaitanaros S, Kyriakides S, Kraynik AM (2012) On the crushing response of random open-cell foams. Int J Solids Struct 49(19–20):2733–2743. https://doi.org/10.1016/j.ijsolstr.2012.03.003

11. Rege A, Aney S, Milow B (2021) Influence of pore-size distributions and pore-wall mechanics on the mechanical behavior of cellular solids like aerogels. Phys Rev E 103(4):043001. https://doi.org/10.1103/PhysRevE.103.043001
12. Rege A (2021) Constitutive modeling of the densification behavior in open-porous cellular solids. Materials 14(11):2731. https://doi.org/10.3390/ma14112731
13. Zhu HX, Hobdell JR, Windle AH (2001) Effects of cell irregularity on the elastic properties of 2d Voronoi honeycombs. J Mech Phys Solids 49:857–870. https://doi.org/10.1016/S0022-5096(00)00046-6
14. Zhu HX, Thorpe SM, Windle AH (2006) The effect of cell irregularity on the high strain compression of 2d Voronoi honeycombs. Int J Solids Struct 43(5):1061–1078. https://doi.org/10.1016/j.ijsolstr.2005.05.008
15. Alsayednoor J, Harrison P, Guo Z (2013) Large strain compressive response of 2-d periodic representative volume element for random foam microstructures. Mech Mater 66:7–20. https://doi.org/10.1016/j.mechmat.2013.06.006
16. Alsayednoor J, Harrison P (2016) Evaluating the performance of microstructure generation algorithms for 2-d foam-like representative volume elements. Mech Mater 98:44–58. https://doi.org/10.1016/j.mechmat.2016.04.001
17. Rege A, Hillgärtner M, Itskov M (2019) Mechanics of biopolymer aerogels based on microstructures generated from 2-d Voronoi tessellations. J Supercrit Fluids 151:24–29. https://doi.org/10.1016/j.supflu.2019.04.018
18. Jang W-Y, Kraynik AM, Kyriakides S (2008) On the microstructure of open-cell foams and its effect on elastic properties. Int J Solids Struct 45(7–8):1845–1875. https://doi.org/10.1016/j.ijsolstr.2007.10.008
19. Fu B, Luo H, Wang F, Churu G, Chu KT, Hanan JC, Sotiriou-Leventis C, Leventis N, Lu H (2011) Simulation of the microstructural evolution of a polymer crosslinked templated silica aerogel under high-strain-rate compression. J Non-Cryst Solids 357(10):2063–2074. https://doi.org/10.1016/j.jnoncrysol.2011.02.049
20. Skibinski J, Cwieka K, Haj Ibrahim S, Wejrzanowski T (2019) Influence of pore size variation on thermal conductivity of open-porous foams. Materials (Basel) 12(12):2485–2494. https://doi.org/10.3390/ma12122017
21. Foray G, Randrianalisoa JH, Adrien J, Maire E (2022) X-ray tomography coupled with finite elements, a fast method to design aerogel composites and prove their superinsulation experimentally. Gels 8(11):732. https://doi.org/10.3390/gels8110732
22. Chandrasekaran R, Hillgärtner M, Ganesan K, Milow B, Itskov M, Rege A (2021) Computational design of biopolymer aerogels and predictive modelling of their nanostructure and mechanical behaviour. Sci Rep 11:10198. https://doi.org/10.1038/s41598-021-89634-1
23. Aney S, Rege A (2023) The effect of pore sizes on the elastic behaviour of open-porous cellular materials. Math Mech Solids 28(7):1624–1634. https://doi.org/10.1177/10812865221124142
24. Miehe C, Schröder J, Schotte J (1999) Computational homogenization analysis in finite plasticity S imulation of texture development in polycrystalline materials. Comput Methods Appl Mech Eng 171(3):387–418. https://doi.org/10.1016/S0045-7825(98)00218-7
25. Smit RJM, Brekelmans WAM, Meijer HEH (1998) Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling. Comput Methods Appl Mech Eng 155(1):181–192. https://doi.org/10.1016/S0045-7825(97)00139-4
26. Schröder J (2014) A numerical two-scale homogenization scheme: the FE^2 -method. In: CISM international centre for mechanical sciences, courses and lectures, vol 550, pp 1–64. https://doi.org/10.1007/978-3-7091-1625-8_1
27. Kouznetsova V, Brekelmans WAM, Baaijens FPT (2001) An approach to micro-macro modeling of heterogeneous materials. Comput Mech 27(1):37–48. https://doi.org/10.1007/s004660000212
28. Feyel F (1999) Multiscale FE^2 elastoviscoplastic analysis of composite structures. Comput Mater Sci 16(1):344–354. https://doi.org/10.1016/S0927-0256(99)00077-4
29. Rege A, Schestakow M, Karadagli I, Ratke L, Itskov M (2016) Micro-mechanical modelling of cellulose aerogels from molten salt hydrates. R Soc Chem 12:7079–7088. https://doi.org/10.1039/C6SM01460G
30. Rege A, Preibisch I, Schestakow M, Ganesan K, Gurikov P, Milow B, Smirnova I, Itskov M (2018) Correlating synthesis parameters to morphological entities: predictive modeling of biopolymer aerogels. Materials 1:1–19. https://doi.org/10.3390/ma11091670
31. Abdusalamov R, Scherdel C, Itskov M, Milow B, Reichenauer G, Rege A (2021) Modeling and simulation of the aggregation and the structural and mechanical properties of silica aerogels. J Phys Chem B 125(7):1944–1950. https://doi.org/10.1021/acs.jpcb.0c10311
32. Somnic J, Jo BW (2022) Status and challenges in homogenization methods for lattice materials. Materials. https://doi.org/10.3390/ma15020605
33. Vigliotti A, Deshpande VS, Pasini D (2014) Non linear constitutive models for lattice materials. J Mech Phys Solids 64:44–60. https://doi.org/10.1016/j.jmps.2013.10.015
34. Vigliotti A, Pasini D (2012) Linear multiscale analysis and finite element validation of stretching and bending dominated lattice materials. Mech Mater 46:57–68. https://doi.org/10.1016/j.mechmat.2011.11.009
35. Vigliotti A, Pasini D (2018) Chapter 8—the design of skin panels for morphing wings in lattice materials. Butterworth-Heinemann, pp 231–246. https://doi.org/10.1016/B978-0-08-100964-2.00008-3
36. Wang XL, Stronge WJ (2001) Micro-polar theory for a periodic force on the edge of elastic honeycomb. Int J Eng Sci 39(7):821–850. https://doi.org/10.1016/S0020-7225(00)00065-3
37. Wang A-J, McDowell DL (2004) In-plane stiffness and yield strength of periodic metal honeycombs. J Eng Mater Technol 126(2):137–156. https://doi.org/10.1115/1.1646165
38. Deshpande VS, Fleck NA, Ashby MF (2001) Effective properties of the octet-truss lattice material. J Mech Phys Solids 49(8):1747–1769. https://doi.org/10.1016/S0022-5096(01)00010-2
39. Masters IG, Evans KE (1996) Models for the elastic deformation of honeycombs. Compos Struct 35(4):403–422. https://doi.org/10.1016/S0263-8223(96)00054-2
40. Koeppe A, Hernandez Padilla CA, Voshage M, Schleifenbaum JH, Markert B (2018) Efficient numerical modeling of 3d-printed lattice-cell structures using neural networks. Manuf Lett 15:147–150. https://doi.org/10.1016/j.mfglet.2018.01.002. Industry 4.0 and Smart Manufacturing
41. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101. https://doi.org/10.1016/j.cma.2016.02.001
42. Conti S, Müller S, Ortiz M (2018) Data-driven problems in elasticity. Arch Ration Mech Anal 229:79–123. https://doi.org/10.1007/s00205-017-1214-0
43. Korzeniowski TF, Weinberg K (2022) Data-driven finite element computation of open-cell foam structures. Comput Methods Appl Mech Eng 400:115487. https://doi.org/10.1016/j.cma.2022.115487
44. Abdusalamov R, Pandit P, Milow B, Itskov M, Rege A (2021) Machine learning-based structure-property predictions in silica aerogels. Soft Matter 17(31):7350–7358. https://doi.org/10.1039/D1SM00307K

45. Pandit P, Abdusalamov R, Itskov M, Rege A (2024) Deep reinforcement learning for microstructural optimisation of silica aerogels. Sci Rep 14(1):1511. https://doi.org/10.1038/s41598-024-51341-y
46. Moulinec C, Houzeaux G, Borrell R, Quintanas Corominas A, Oyarzun G, Grasset J, Giuntoli G, Vazquez M (2023) A massively parallel multi-scale FE2 framework for multi-trillion degrees of freedom simulations. In: Proceedings of the platform for advanced scientific computing conference. PASC '23. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3592979.3593415
47. Klawonn A, Lanser M, Rheinbach O, Uran M (2021) Fully-coupled micro-macro finite element simulations of the Nakajima test using parallel computational homogenization. Comput Mech 68(5):1153–1178. https://doi.org/10.1007/s00466-021-02063-9
48. Klawonn A, Köhler S, Lanser M, Rheinbach O (2020) Computational homogenization with million-way parallelism using domain decomposition methods. Comput Mech 65(1):1–22. https://doi.org/10.1007/s00466-019-01749-5
49. Le BA, Yvonnet J, He Q-C (2015) Computational homogenization of nonlinear elastic materials using neural networks. Int J Numer Meth Eng 104(12):1061–1084. https://doi.org/10.1002/nme.4953
50. Fritzen F, Fernandez M, Larsson F (2019) On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling. Front Mater. https://doi.org/10.3389/fmats.2019.00075
51. Gupta A, Bhaduri A, Graham-Brady L (2023) Accelerated multiscale mechanics modeling in a deep learning framework. Mech Mater 184:104709. https://doi.org/10.1016/j.mechmat.2023.104709
52. Eidel B (2023) Deep CNNs as universal predictors of elasticity tensors in homogenization. Comput Methods Appl Mech Eng 403:115741. https://doi.org/10.1016/j.cma.2022.115741
53. Gajek S, Schneider M, Böhlke T (2021) An FE-DMN method for the multiscale analysis of short fiber reinforced plastic components. Comput Methods Appl Mech Eng 384:113952. https://doi.org/10.1016/j.cma.2021.113952
54. Rocha IBCM, Kerfriden P, van der Meer FP (2021) On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. J Comput Phys X 9:100083. https://doi.org/10.1016/j.jcpx.2020.100083
55. Eivazi H, Tröger J-A, Wittek S, Hartmann S, Rausch A (2023) FE2 computations with deep neural networks: algorithmic structure, data generation, and implementation. Math Comput Appl. https://doi.org/10.3390/mca28040091
56. Rocha IBCM, Kerfriden P, van der Meer FP (2023) Machine learning of evolving physics-based material models for multiscale solid mechanics. Mech Mater 184:104707. https://doi.org/10.1016/j.mechmat.2023.104707
57. Feng N, Zhang G, Khandelwal K (2022) Finite strain FE2 analysis with data-driven homogenization using deep neural networks. Comput Struct 263:106742. https://doi.org/10.1016/j.compstruc.2022.106742
58. Niekamp R, Niemann J, Schröder J (2023) A surrogate model for the prediction of permeabilities and flow through porous media: a machine learning approach based on stochastic Brownian motion. Comput Mech 71(3):563–581. https://doi.org/10.1007/s00466-022-02250-2
59. Deng S, Hosseinmardi S, Wang L, Apelian D, Bostanabad R (2024) Data-driven physics-constrained recurrent neural networks for multiscale damagemodeling of metallic alloys with process-induced porosity. Comput Mech. https://doi.org/10.1007/s00466-023-02429-1
60. Gajek S, Schneider M, Böhlke T (2021) An FE-DMN method for the multiscale analysis of thermodynamical composites. Comput Mech 69:1087–1113. https://doi.org/10.1007/s00466-021-02131-0
61. Broyden CG (1970) The convergence of a class of double-rank minimization algorithms: 2. The new algorithm. IMA J Appl Math 6(3):222–231. https://doi.org/10.1093/imamat/6.3.222
62. Öchsner A (2023) Euler–Bernoulli beams and frames. Springer, Cham, pp 103–226. https://doi.org/10.1007/978-3-031-09673-0_3
63. Kassimali A (2011) Matrix analysis of structures, 2nd edn. Cengage Learning. https://www.academia.edu/36080840/Matrix_Analysis_of_Structures_2nd_ed
64. Becker R, Sokolow A (2015) Stress averaging for a beam network for use in hierachical multiscale framework. ARL-MR-0887. https://apps.dtic.mil/sti/citations/ADA619976
65. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TnsorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. https://www.tensorflow.org/
66. Aney S, Pandit P, Ratke L, Milow B, Rege A (2023) On the origin of power-scaling exponents in silica aerogels. J Sol-Gel Sci Technol. https://doi.org/10.1007/s10971-023-06156-0
67. Hendrycks D, Gimpel K (2016) Gaussian error linear units (GELUs). https://doi.org/10.48550/arXiv.1606.08415
68. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980
69. Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York. https://doi.org/10.1007/978-0-387-40065-5
70. Byrd RH, Nocedal J, Yuan Y-X (1987) Global convergence of a class of quasi-newton methods on convex problems. SIAM J Numer Anal 24(5):1171–1190. https://doi.org/10.1137/0724077
71. Dai Y-H (2002) Convergence properties of the bfgs algoritm. SIAM J Optim 13(3):693–701. https://doi.org/10.1137/S1052623401383455
72. Wolfe P (1969) Convergence conditions for ascent methods. SIAM Rev 11(2):226–235. https://doi.org/10.1137/1011036

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.