Applying neural networks to analyse the properties and structure of composite materials

A. Valishin[1*] and *N.* Beriachvili[1]

[1]Bauman Moscow State Technical University, 2nd Baumanskaya st., 5/1, 105005 Moscow, Russia

Abstract. In this paper, we show how a simple convolutional neural network (CNN) can be trained to predict homogenised elastic properties of a composite material based on its representative volume element (RVE). We consider both 2D and 3D composites featuring two components with fixed isotropic elastic properties. The dataset used to train the neural network is obtained in two stages. The first stage is the generation of random RVEs and computation of their homogenised elastic properties using finite element analysis (FEA) with periodic boundary conditions (PBC) applied to them, and the second stage is the application of a data augmentation scheme to that dataset obtained in the first stage. For the 2D and 3D cases, we present two neural networks and show that they are able to estimate the values of homogenised elastic properties fairly accurately. At the same time, we confirm that the data generation stage is computationally very expensive and is a major challenge for machine learning based techniques in computational mechanics. Indeed, it is expensive even for "low resolution" RVEs such as those considered in our work, i.e. squares (2D case) or cubes (3D case) uniformly subdivided into 8x8 (2D case) or 8x8x8 (3D case) domains. We discuss the flaws and drawbacks as well as the possible developments and uses of this approach in multiscale modeling of composite materials. Key words: composite materials, asymptotic homogenisation method, numerical simulations, finite element analysis, multiscale modeling, artificial intelligence, convolutional neural networks

1 Introduction

In recent years, thanks to large amounts of available data and computing power, data-driven approaches and AI based on machine learning techniques have spread to engineering fields, including composite materials science. Artificial neural networks (ANN) models are arguably the most popular machine learning technique due to their great performance in many fields and their ability to approximate very complex relations. Some of the most interesting opp5rtunities brought by AI to composite materials science and mechanics include discovery of unknown constitutive laws, acceleration of multiscale modeling, design optimisation, damage detection and quality control integrated with manufacturing processes, additive manufacturing, and analysis of process-structure-property-performance relationships [1-9]. However, despite the promising uses of AI in composite materials mechanics, new challenges

* Corresponding author: enf@mail.ru

and problems arise, like the high computational cost of high-quality data generation based on complex numerical simulations and mechanical models [1-9].

In this paper, we present our work on training a CNN to predict homogenised elastic orthotropic properties of composites with two fixed components based on their RVE. In the rest of the paper we describe the problem statement, the data generation and augmentation stages and the architecture of the neural networks in the 2D and 3D cases. We then demonstrate our results and evaluate the presented neural networks and their training. Finally, we discuss the flaws and drawbacks, the possible developments and uses of this approach in multiscale modeling of composite materials.

2 Problem statement

The main objective is to predict the homogenised elastic oprthotropic properties of a composite based on its RVE. In the rest of the paper, we consider the following assumptions:

- The considered composites have only two components (matrix and filler) with fixed, isotropic and elastic properties (see Table 1).
- The contact at the matrix-filler interfaces is ideal (total adhesion, no friction).
- The filler volume fraction is situated between 0.2 and 0.8.
- The composite has a periodic structure : as such the RVE is considered to be a periodicity cell.
- The RVEs are modeled as squares (2D case) or cubes (3D case) with a 20µm side length and uniformly subdivided into 8x8 square domains (2D case) or 8x8x8 cubic domains (3D case) with every domain being assigned a component material (either matrix or filler).

Table 1. The components' elastic properties values.

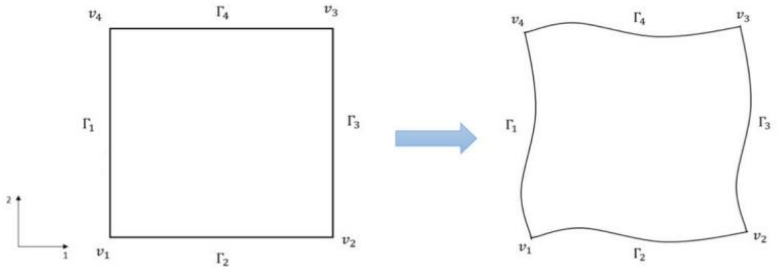
Component	Young Modulus (E)	Poisson Ratio (ν)
Matrix	0.1 GPa	0.33
Filler	1 GPa	0.33

In multiscale modeling the asymptotic homogenisation method [10,11] consists of prescribing unitary average strains (or respectively stresses) to the RVE under periodic boundary conditions (Figure 2) and averaging the stress (or respectively strain) in the RVE to obtain a "macroscopic" (total average) stress (or respectively strain). In turn, this provides access to a corresponding average (homogenised) stiffness matrix and its inverse, the homogenised compliance matrix. In our case, the components (filler and matrix) are isotropic and the RVE is square (or cubic), and so it seems appropriate to choose an orthotropic homogenisation of the composites. Thus, in the homogenised orthotropic compliance matrix (Figure 1) we can extract the sought-after homogenised elastic properties (E_1, E_2, G_{12} in the 2D case, E_1, E_2, E_3, G_{12}, G_{23}, G_{13} in the 3D case) .

It should be noted we could have also considered ν_{12}, ν_{21}, ν_{13}, ν_{31}, ν_{23}, ν_{32}, or even all of the matrix' coefficients, which would have made sense if we were considering a more general (anisotropic) homogenisation. However, in the beginning of our work we chose to consider a limited amount of coefficients for simplicity.

$$\underline{\underline{S}} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{31}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix}$$

Fig. 1. Compliance matrix of an orthotropic material (3D case).



$$u_{dst} - u_{src} = \underline{\varepsilon}(X_{dst} - X_{src})$$

Fig. 2. (from [9]) Periodic boundary conditions (2D case), (src, dst) pairs - (r1, r3), (r2, r4), u: displacement field, X: material coordinates, $\underline{\varepsilon}$: strain tensor.

Using software capable of modelling RVEs (subject to the assumptions listed above) and performing FEA, we can compute the homogenised compliance matrix and extract our sought-after properties. As such, we decide to generate a large amount of random RVEs, compute their homogenised elastic properties to form a dataset and use it to train a neural network for a regression task - the prediction of homogenised elastic properties for a priori unknown input RVEs.

3 Methodology

3.1 FEA and data generation

To model the RVEs and apply the homogenisation method to them we used COMSOL Multiphysics 5.6 and the Cell Periodicity node of its Solid Mechanics physics module [12,13]. Additionally, we used COMSOL MATLAB LiveLink to automate the operations in COMSOL and export the results of its computations. As such, we wrote MATLAB code to assign the filler material to random domains of the RVEs, compute the associated homogenised elastic properties, and save them in a table (.csv file) with the file names of the corresponding RVEs, which are saved as well in .mat files. The RVEs are represented in .mat files as arrays whose values represent the material assigned to the domain at the corresponding position (0 for matrix, 1 for filler).

Using this code, we generated datasets of 10000 (2D case) and 1000 (3D case) random RVEs (with filler volume fractions evenly distributed between 0.2 and 0.8) and their corresponding homogenised elastic properties. It took about 5 hours to generate the 10000 2D samples and about 11 hours to generate the 1000 3D ones. Finally, each of the two datasets were split into training (80%), validation (10%) and test (10%) sets.

Additionally, the training datasets are subject to a data augmentation scheme described in [14], relying on simple geometric transformations of the RVE and, additionally, its periodicity (Figure 3). Data augmentation schemes have a regularisation effect on neural networks and can reduce overfitting and improve generalisation. However, relying on them too much can affect generalisation too, in particular when the data distribution of the initial and augmented datasets are significantly different.

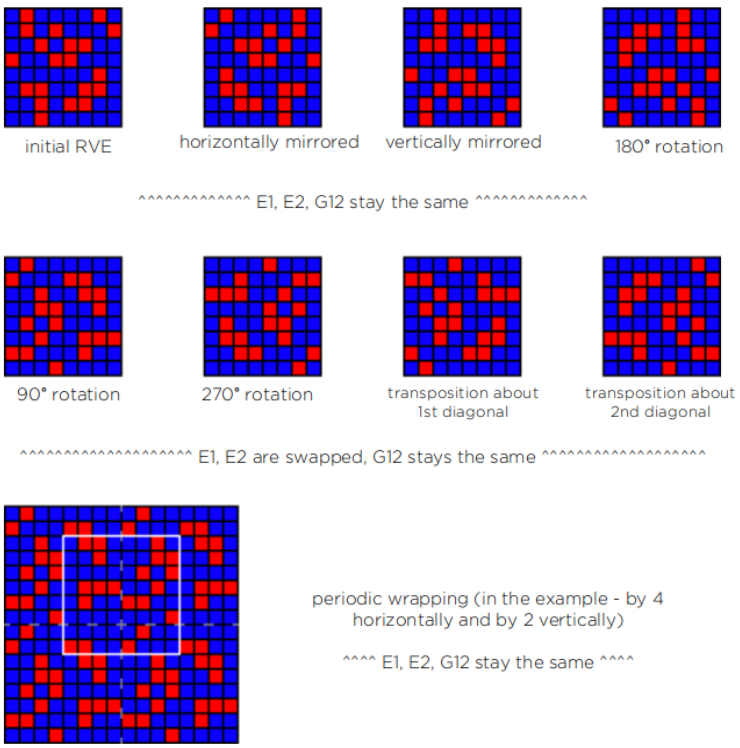


Fig. 3. The new RVEs and associated homogenised elastic properties obtained using the data augmentation scheme on an initial RVE. (2D case example. 3D case is analogous).

In theory, this data augmentation scheme allows to increase the amount of samples in the training set from 8000 to 568000 (71 times more) in the 2D case, and from 800 to 424000 (530 times more) in the 3D case. However in the 3D case, we eventually used only a partial augmentation scheme, where we skipped one in every two domains when iterating through the RVE's domains in a single direction. This was done to prevent too many very similar RVEs with identical associated properties from appearing in the new dataset, making it much less diverse. Indeed, the proportion of new and a priori less similar RVEs generated from other geometric transformations (which contributes 18 times more samples) could become too insignificant compared to those generated from periodic wrapping (which contributes 511 times more samples in the full version, and 45 times more in the partial version). With the

partial augmentation scheme, the amount of samples in training dataset in the 3D case increased from 800 to 51200 (64 times more).

3.2 Neural network architecture and training

In our work we used Python 3.8 and the Tensorflow and Keras libraries to implement the neural networks. To manage datasets the Pandas library was partially used as well.

Because the composites' RVEs discretised, they are essentially visual representations or "images", convolutional neural networks seem to be the appropriate machine learning method for our regression problem (prediction of homogenised elastic properties). The architecture and hyperparameters are presented in Table 2.1 and Table 2.2 for the 2D case, and Table 3.1 and Table 3.2 for the 3D case. As a reminder, there are only 3 sought-after values in the 2D case (E_1, E_2 and G_{12}) which is why there are only 3 neurons in the final layer in Table 2, compared to the 6 final neurons of Table 5.

Table 2. Neural network architecture (2D case).

Input layer (input shape: 8x8)
Convolutional layer (Conv2D) (filters: 32@8x8, kernel size: 3x3, padding: SAME)
Batch normalization layer
ReLU layer (Rectified linear unit)
Convolutional layer (Conv2D) (filters: 64@8x8, kernel size: 3x3, padding: SAME)
Пакетная нормализация (BatchNormalization)
Batch normalization layer
Flatten layer
Dense layer (64 neurons, ReLU activation)
Dense layer (128 neurons, ReLU activation)
Dense layer (64 neurons, ReLU activation)
Dense layer (3 neurons)

Table 3. Neural network hyperparameters (2D case).

Loss function : MSE (Mean squared error)
Optimiser : ADAM
Learning rate : 0,0001
Batch size : 32
Maximum epochs : 32
Early stopping : 5

Table 4. Neural network architecture (3D case).

Input layer (input shape: 8x8x8)
Convolutional layer (Conv3D) (filters: 256@8x8x8, kernel size: 3x3x3, padding: SAME)
Batch normalization
Pooling layer (MaxPool3D) (size: 2x2x2, padding: SAME)
ReLU layer
Convolutional layer (Conv3D) (filters: 512@8x8x8, kernel size: 3x3x3, padding: SAME)
Batch normalization
Pooling layer (MaxPool3D) (size: 2x2x2, SAME padding)
ReLU layer
Flatten layer
Dense layer (128 neurons, ReLU activation)
Dense layer (256 neurons, ReLU activation)
Dense layer (256 neurons, ReLU activation)

Dense layer (512 neurons, ReLU activation)
Dense layer (512 neurons, ReLU activation)
Dense layer (256 neurons, ReLU activation)
Dense layer (256 neurons, ReLU activation)
Dense layer (128 neurons, ReLU activation)
Dense layer (6 neurons)

Table 5. Neural network hyperparameters (3D case).

Loss function : MSE (Mean squared error)
Optimiser : ADAM
Learning rate : 0,00001
Batch size : 32
Maximum epochs : 500
Early stopping : 10

It should be noted that the "resolution" of the RVEs in the 2D case is very low (8x8). This does not advocate for the use of a CNN, as pooling layers are inappropriate for such low resolutions. As such, there is no pooling layers in Table 2. At such a low resolution, the neural network isn't much different from a standard dense neural network with a input layer of 64 neurons. But at the same time, in the 3D case or in the case of a higher resolution 2D RVE, a standard dense neural network wouldn't be appropriate anymore, as the "feature extraction" capability of CNNs would be very desirable. As such, in the 3D case (Table 5) despite a similarly low resolution (8x8x8) pooling layers are actually present.

Finally, the targets of the regression had to be adjusted, and so the values of the homogenised elastic properties in the datasets were divided by 10^8. This was necessary for the correct behavior of the optimiser of choice, based on stochastic gradient descent (ADAM [15]), as it was unable to make the loss function converge before the adjustment. Indeed, the loss funtion was stalling at very high values (10^8 order of magnitude) and was decreasing extremely slowly.

4 Results

All work was done on computer with an Intel Core i7-7700HQ@2.80GHz x8 CPU and 8 Gb of RAM. No GPU acceleration was used for the neural network training.

The training results are presented in Table 6 for the 2D case and Table 7 for the 3D case. In both cases the data augmentation scheme has a significant positive effect on the overall prediction accuracy, evaluated using the mean absolute percentage error (MAPE) on the test set. But as expected, the training times are far longer with augmented datasets. However, these training times are far shorter then the data generation durations. As a reminder, generating 10000 2D samples took around 5 hours, and generating 1000 3D samples took more than 11 hours.

Table 6. Evaluation metrics for the neural network and its training (2D case).

	Without data augmentation	With data augmentation
Training set size	8000	568000
Final training epoch	241	43
Training time	5 minutes	50 minutes
Test set loss	0.0186	0.0024
Test set MAPE	4.0929%	1.4965%

Table 7. Evaluation metrics for the neural network and its training (3D case).

	Without data augmentation	With data augmentation
Training set size	800	51200
Final training epoch	87	29
Training time	38 minutes	5 hours 48 minutes
Test set loss	0.0236	0.0048
Test set MAPE	4.3632%	1.9673%

Figure 5, Figure 6, Figure 7 and Figure 8 show comparisons between ground truth and predicted homogenised elastic properties for the test set, in both 2D and 3D cases, with and without augmented training data. Predicted values appear to be close to ground truth values, and the variance is much lower in the case of models that were trained on augmented datasets. However, both in the augmented and non-augmented cases, the variance gets higher as the ground truth values get larger.

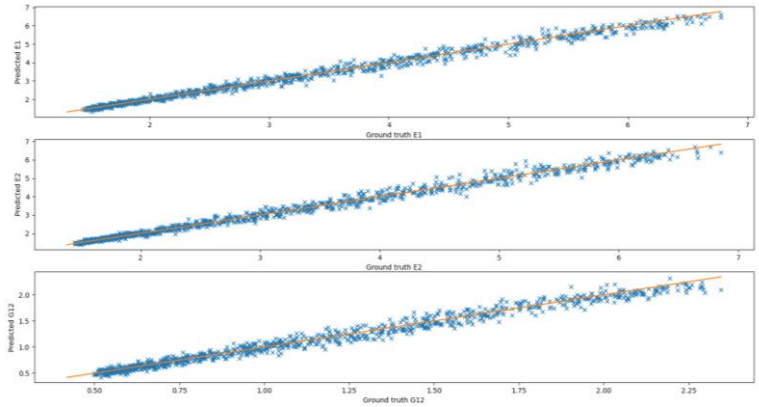


Fig. 5. Comparison betweeen ground truth and predicted homogenised elastic properties. ($1/10^8$ ratio) for the test set (2D case, no data augmentation).

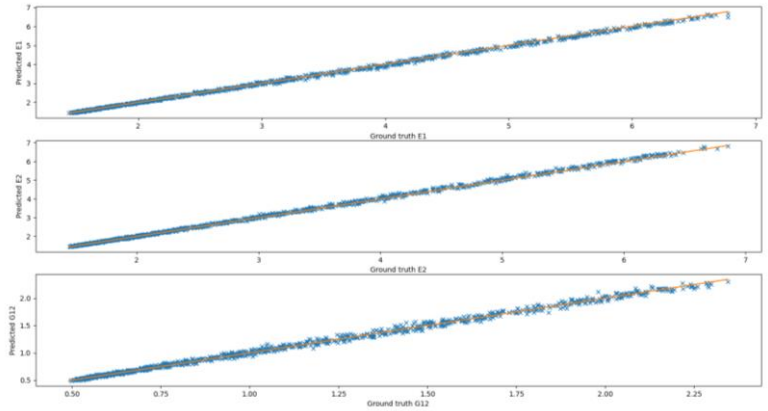


Fig. 6. Comparison betweeen ground truth and predicted homogenised elastic properties ($1/10^8$ ratio) for the test set (2D case, with data augmentation).

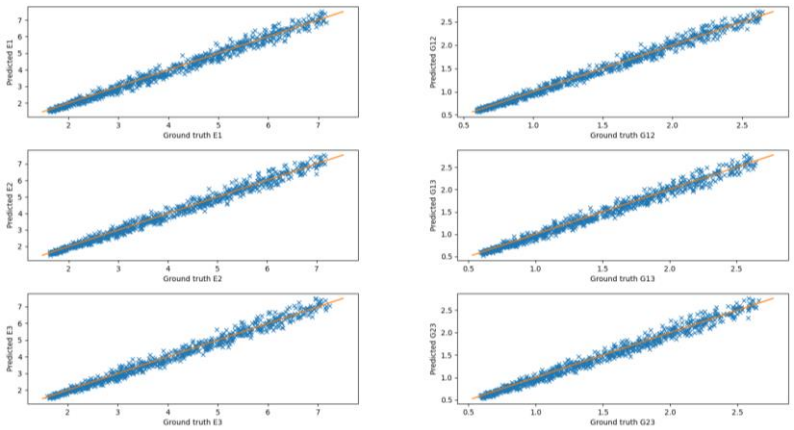


Fig. 7. Comparison betweeen ground truth and predicted homogenised elastic properties ($1/10^8$ ratio) for the test set (3D case, no data augmentation).

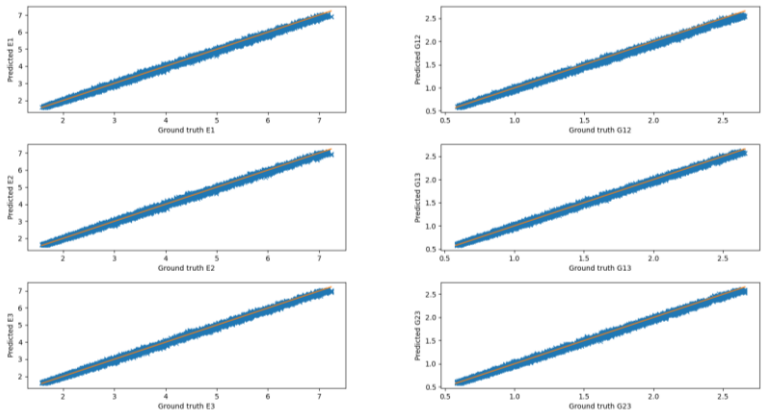


Fig. 8. Comparison betweeen ground truth and predicted homogenised elastic properties ($1/10^8$ ratio) for the test set (3D case, with data augmentation).

5 Discussion and further perspectives

The results of our work confirm that data-driven approaches such as the one presented here are indeed capable of quickly returning estimated values for mechanical properties, which, for example, can be useful at some prototyping stages in the modeling or the design of a composite. As a matter of fact, despite having lower accuracy than a traditional numerical simulation method (like FEA) the neural network is significantly faster, as it returns a result almost instantly (less than a second) while traditional FEA on an RVE can take a few seconds.

This approach, which was initially applied to the 2D case was successfully extended to the 3D case as well, at the cost of a slightly more complex neural network architecture and much longer data generation and training time.

Data generation is indeed the most computationally expensive stage in the making of this predictive system, as conducting FEA on a large amount of random RVEs ends up being very slow, even compared to the neural network training. As such, one of the most crucial challenges in data-driven approaches to mechanics is the reduction of the computational cost of data generation. The data augmentation scheme used in our work helps in the creation of

the final dataset, but it is far from being enough, and more effective and smart ways to generate and use data in training are very much needed.

The RVE representation used in our work (8x8 squares and 8x8x8 cubes) isn't really appropriate for data generation (using FEA), as it involves additional finite elements on interfaces between domains assigned to the same component material. Moreover, the resolution of this discretised representation (8x8, 8x8x8) is very low and severely limits the diversity of possible RVEs. As such, simpler and more appropriate RVE representations should be used for numerical simulations (data generation) before being converted to a discretised visual representation for learning stages, as it is suggested in [16].

The approach showcased in this paper has a lot of space for improvements but also good potential, which may be explored in future work.

A very interesting new feature would be the support for user-defined amounts of component materials (not just two) with user-defined material properties, as it would be very desirable for fast modeling or design prototyping needs. Moreover, there is also great interest in the extension of the approach to other mechanical properties of composites, related to other mechanical theories, such as fracture mechanics.

Another interesting promise for this predictive system is design optimisation, with the development of inverse generative models, capable of suggesting a composite structure (RVE) to satisfy constraints on maximum, minimum, or user-defined values for homogenised properties, filler volume fraction etc. Such a generative model is presented in [14], and [17] pursues a similar goal using a genetic algorithm.

Finally, it is worth exploring a more hybrid variant of this approach, similar to the model-data-driven one presented in [18].

6 Conclusion

In this paper, we showed how a convolution neural network can be used to predict homogenised orthotropic elastic properties of composites featuring two components with fixed isotropic elastic properties, based on their RVE, discretised as 8x8 squares in the 2D case and 8x8x8 cubes in the 3D case. The general assumptions were presented in the problem statement, as well as the methodology of the work, consisting of data generation, data augmentation, and neural network training stages. The architecture, training results, and performance evaluation of the neural networks both in the 2D and 3D cases was showed as well. Finally, we discussed the flaws, possible improvements and perspectives of this data-driven computational mechanics approach.

References

1. YU.I. Dimitriyenko, *Nelineynaya mekhanika sploshnoy sredy* (Fizmatlit, M., 2009)
2. YU.I. Dimitriyenko, *Mekhanika sploshnoy sredy: uch. posobiye v 4-kh tomakh. Tom 4: Osnovy mekhaniki tvordykh sred* (Iz-vo MGTU im. N.E. Baumana, Moskva, 2013)
3. YU.I. Dimitriyenko, YU.V. Yurin, S.V. Sborshikov, A.D. Yakhnovskiy, R.R. Baymurzin, Matematicheskoye modelirovaniye i chislennyye metody **3**, 22–46 (2020)
4. YU.I. Dimitriyenko, YE.A. Gubareva, D.O. Yakovlev, Nauchnoye izdaniye MGTU im. N.E. Baumana **10**, 359–382 (2014)
5. YU.I. Dimitriyenko, YE.A. Gubareva, S.V. Sborshikov, Matematicheskoye modelirova-niye i chislennyye metody **2**, 28–48 (2014)
6. X. Liu, S. Tian, F. Tao, W. Yu, Composites Part B: Engineering **224**, 109152 (2021)

7. X. Liu, S. Tian, F. Tao, W. Yu, H*ow machine learning can help the design and analysis of composite materials and structures? arXiv preprint arXiv:2010.09438* (2021)
8. J.J. De Pablo, N.E. Jackson, M.A. Webb et al., Computational Materials **5**, 1-23 (2019)
9. M.R.P. Elenchenzhian, V. Vadlamudi, R. Raihan et al., *Data driven composites: the challenge and paths forwards. Proceedings of CAMX Conference* (2018)
10. N.S. Bakhvalov, G.N. Panasenko, *Homogenisation*, *Averaging Processes in Periodic Media* (Springer, Dordrecht, 1989)
11. G. Papanicolau, A. Bensoussan, J.L. Lions, *Asymptotic analysis for periodic structures* (Elsevier, 1978)
12. *Cell Periodicity*, URL: https://doc.comsol.com/5.5/doc/com.comsol.help.sme/sme_ug_solid.07.70.html
13. *Periodic Cell Theory*, URL: *https://doc.comsol.com/5.5/doc/com.comsol.help.sme/sme_ug_theory.06.63.html*
14. S. Chakravarty, P. Garg, A. Kumar et al., *Deep neural networks based predictive-generative framework for designing composite materials. arXiv preprint arXiv:2105.01384* (2021)
15. D.P. Kingma, J. Ba, *Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980* (2014)
16. C. Rao, Y. Liu, Computational Materials Science **184**, 109850 (2020)
17. D.W. Abueidda, M. Almasri, R. Ammourah et al., Composite Structures **227**, 111264 (2019)
18. J.N. Fuhg, C. Böhm, N. Bouklas et al., *Model-data-driven constitutive responses: application to a multiscale computational framework. arXiv preprint arXiv:2104.02650* (2021)