Noname manuscript No.
(will be inserted by the editor)

Enhancing the accuracy of deep learning based FE^2 algorithm using proper orthogonal decomposition

Saumik Dana

Received: date / Accepted: date

Abstract In the earlier work, we proposed a machine learning leveraged FE^2 algorithm for two-scale modeling of elastic solids. The method eliminates the need to solve the RVE problem on-the-fly by replacing the effective input-output causality by a neural network. This potentially reduces the computational cost of the FE^2 algorithm significantly. In this work, we put forth the use of snapshot proper orthogonal decomposition to improve the accuracy of the machine learning leveraged algorithm. Instead of training one neural net, multiple neural nets are trained with the coefficients of the basis of the snapshot matrix as the target.

Keywords Machine learning · FE^2 homogenization · Proper orthogonal decomposition

1 Introduction

The RVE concept [1–6] is commonly used in the manufacturing sector to avoid using computationally expensive simulation platforms necessary to capture microstructural features. In essence, the features are captured in the RVE and averaged out over the RVE before any discretization technique is employed at the macroscale with the averaged properties as parameters. The popular FE^2 algorithm [7–9] is commonly employed in which each gauss point for the finite element calculations at the macroscale is associated with a RVE and the information exchange between the two scales occurs at each of those gauss points. A 2D depiction of the algorithmic framework for elastic solids is given in Figure 1. The concepts of deformation gradient and the particular stress measure are explained in Appendix A. The deformation gradient manifests itself as periodic boundary conditions on the RVE as explained in Appendix B. The macroscale first P-K stress is obtained from the RVE scale finite element solution as explained next. Let N be the number of boundary nodes for the RVE scale discretized domain, let $\mathbf{f}_i$ be the force

S. Dana
University of Southern California, CA 90007
E-mail: sdana@usc.edu

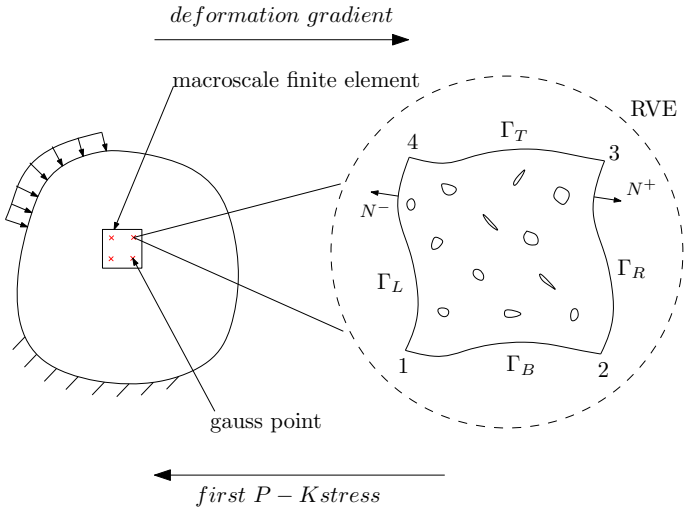

Fig. 1 A 2D depiction of the FE^2 framework for elastic solids. The macroscale problem is discretized into finite elements. The gauss point level computations work in conjunction with RVE scale finite element solve corresponding to each gauss point. The information fed by the macroscale to the RVE is the deformation gradient and the information fed by the RVE to the macroscale is the homogenized stress measure.

on i^{th} boundary node and $\mathbf{X}_i$ be the coordinate of the i^{th} boundary node. The macroscale first P-K stress is obtained as

$$\mathbf{P}_M = \frac{1}{V_0}\sum_{i=1}^{N} \mathbf{f}_i \otimes \mathbf{X}_i \tag{1}$$

where V_0 is the area of the RVE. The derivation of (1) is given in Appendix D.

1.1 A "naive" neural network

The information exchange would need to occure multiple times at every increment of macro load to satisfy convergence. Depending on the complexity of the microstructure, the RVE solve itself would entail a lot of finite elements to resolve all the features in the RVE. The cumbersome computational cost would make the algorithm infeasible for complex microstructures. In lieu of that, [6] designed a machine learning leveraged FE^2 algorithm which eliminates the need for on-the-fly finite element solves at the RVE scale. This is achieved by following the outline laid out in [10] to build a neural net input-output causality relationship for the RVE thereby rendering the RVE as a blackbox which takes the macroscale deformation gradient as input and spits out the macroscale first P-K stress as output as shown in Figure 2. A basic understanding of neural networks in given in Appendix C. Consider the vector of boundary forces $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N)^T$ over a finite element mesh where N is number of boundary nodes. The steps in the machine learning addendum to the algorithm are

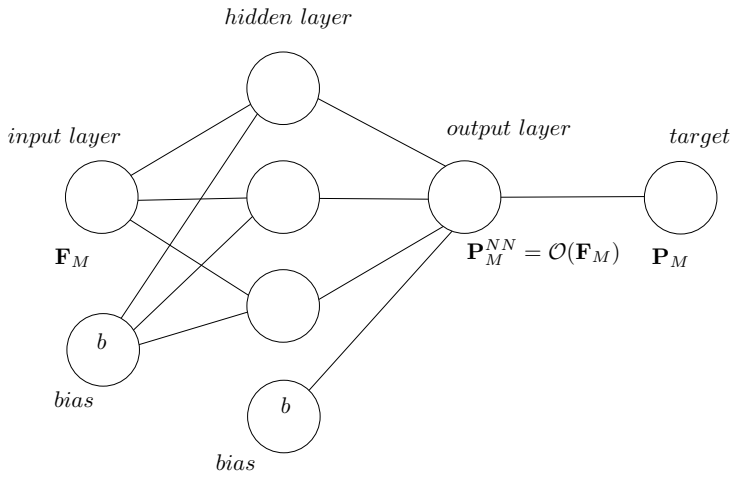

Fig. 2 A naive 1-3-1 neural network with macroscale deformation gradient as input and homogenized first P-K stress as the target output.

– Multiple snapshots of $\mathbf{f}$ are taken over a range of deformation gradients as follows

$$\{\mathbf{f}^{(k)}, k = 1, \cdots, M\}$$

with each snapshot generated from the finite element solution of the RVE BVP with periodic boundary conditions corresponding to each of the M deformation gradients. The non-linear neo-Hookean model is the stress-strain relation for the finite element simulations

$$\boldsymbol{\sigma} = \frac{1}{2}\frac{\lambda}{J}(J^2 - 1)\mathbf{I} + \frac{\mu}{J}(\mathbf{b} - \mathbf{I})$$

where λ and μ are Lame parameters and $\mathbf{b}$ is left Cauchy Green strain tensor. The homogenized first P-K stress is obtained for each of these deformation gradients using the relationship (1).

– This data is then used to build the input-output causality as follows

$$\mathbf{P}_M^{NN} = \mathcal{O}(\mathbf{F}_M) \tag{2}$$

where $\mathcal{O}$ is the map between the deformation gradient and the output of the neural network.

2 A "better" neural network

Equation (1) clearly shows that the macroscale P-K stress is obtained from the RVE scale finite element solve through the forces obtained on the boundary nodes of the discretization. The boundary forces are summed and effectively averaged over the boundary to obtain the homogenized first P-K stress. As a result, a neural network with the homogenized first P-K stress as the target would only capture the volume average of the RVE scale finite element solution features. With that in mind, a neural net is to be designed with the boundary forces as targets as shown in Figure 3. The steps in the machine learning addendum to the algorithm are

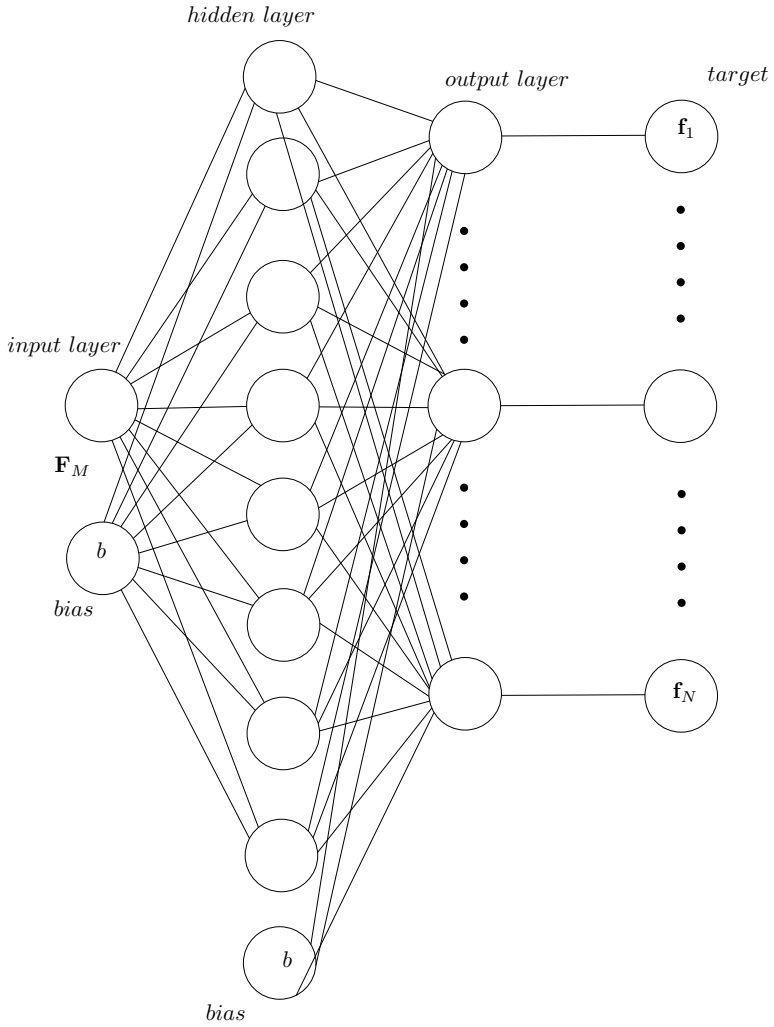

Fig. 3 A neural network with macroscale deformation gradient as input and forces at N boundary nodes as the target output.

– Multiple snapshots of $\mathbf{f}$ are taken over a range of deformation gradients as follows

$$\{\mathbf{f}^{(k)}, k = 1, \cdots, M\}$$

with each snapshot generated from the finite element solution of the RVE BVP with periodic boundary conditions corresponding to each of the M deformation gradients. The non-linear neo-Hookean model is the stress-strain relation for the finite element simulations

$$\boldsymbol{\sigma} = \frac{1}{2}\frac{\lambda}{J}(J^2 - 1)\mathbf{I} + \frac{\mu}{J}(\mathbf{b} - \mathbf{I})$$

where λ and μ are Lame parameters and $\mathbf{b}$ is left Cauchy Green strain tensor.

– This data is then used to build the input-output causality as follows

$$\mathbf{f}^{NN} = \mathcal{O}(\mathbf{F}_M) \tag{3}$$

where $\mathcal{O}$ is the map between the deformation gradient and the output of the neural network.

2.1 Enhancing the accuracy and speed using proper orthogonal decomposition

The deviation matrix is built

$$\mathscr{F} = \begin{bmatrix} \mathbf{f}_1^{(1)} - \bar{\mathbf{f}}_1 & \cdots & \mathbf{f}_1^{(M)} - \bar{\mathbf{f}}_1 \\ . & \cdots & . \\ \mathbf{f}_N^{(1)} - \bar{\mathbf{f}}_N & \cdots & \mathbf{f}_N^{(M)} - \bar{\mathbf{f}}_N \end{bmatrix}_{3N \times M}$$

where $\bar{\mathbf{f}}_j = \frac{1}{M} \sum_{i=1}^{M} \mathbf{f}_j^{(i)}, j = 1, \cdots, N$. The SVD of $\mathscr{F}$ is

$$\mathscr{F} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

The proper orthogonal decomposition ([11]) states that any of the snapshots can be reconstructed by a linear combination of the basis

$$\mathbf{f}^{(k)} = \bar{\mathbf{f}} + \sum_{i=1}^{M} \boldsymbol{\alpha}_i^{(k)} \mathbf{U}_i = \bar{\mathbf{f}} + \boldsymbol{\phi}\boldsymbol{\alpha}^{(k)} \tag{4}$$

where $\bar{\mathbf{f}} = \frac{1}{M}(\sum_{i=1}^{M} \mathbf{f}_1^{(i)}, \cdots, \sum_{i=1}^{M} \mathbf{f}_N^{(i)})^T$, $\mathbf{U}_i$ is the i^{th} column of $\mathbf{U}$ and

$$\boldsymbol{\phi} = \begin{bmatrix} \mathbf{U}_1 \cdots \mathbf{U}_M \end{bmatrix} \qquad , \boldsymbol{\alpha}^{(k)} = (\boldsymbol{\alpha}_1^{(k)}, \cdots, \boldsymbol{\alpha}_M^{(k)})^T = \boldsymbol{\phi}^T \mathscr{F}_k$$

where $\mathscr{F}_k$ is the k^{th} column of $\mathscr{F}$. Instead of one neural net with $\mathbf{f}$ as target, multiple neural nets are trained with $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}$ and so on as target. The first $m < M$ dominant modes can be taken. The reason for increased speed of training is that the neural net training optimizer works on a "local" error defined over each of the deformation gradients with the POD incorporated as opposed to a "global" error defined over all the sampled deformation gradients combined in the absence of POD. The steps in the machine learning addendum to the algorithm are

– Multiple snapshots of $\mathbf{f}$ are taken over a range of deformation gradients as follows

$$\{\mathbf{f}^{(k)}, k = 1, \cdots, M\}$$

with each snapshot generated from the finite element solution of the RVE BVP with periodic boundary conditions corresponding to each of the M deformation gradients. The non-linear neo-Hookean model is the stress-strain relation for the finite element simulations

$$\boldsymbol{\sigma} = \frac{1}{2}\frac{\lambda}{J}(J^2 - 1)\mathbf{I} + \frac{\mu}{J}(\mathbf{b} - \mathbf{I})$$

where λ and μ are Lame parameters and $\mathbf{b}$ is left Cauchy Green strain tensor.

- $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}$ and so on are obtained using POD
- This data is then used to build the input-output causality as follows

$$\begin{cases} \boldsymbol{\alpha}^{(1)^{NN}} = \mathcal{O}_1(\mathbf{F}_M) \\ \boldsymbol{\alpha}^{(2)^{NN}} = \mathcal{O}_2(\mathbf{F}_M) \\ \quad . \\ \quad . \\ \boldsymbol{\alpha}^{(m)^{NN}} = \mathcal{O}_m(\mathbf{F}_M) \end{cases} \tag{5}$$

where $\mathcal{O}_1, \mathcal{O}_2$ and so on are the maps between the deformation gradient and the outputs of the neural network.

2.2 Algorithmic framework in a nutshell

In the initialization stage,

- Establish the input-output causality of the RVE problem as in (5), (4) and (1)

Once the initalization phase is complete,

- An increment of macro load is applied
- Macroscale BVP is solved with macroscale stiffness given by (13)
- The macroscale deformation gradient is updated
- Periodic boundary conditions are imposed on RVE in accordance with (6)
- Homogenized first P-K stress is obtained in accordance with (5), (4) and (1)
- The gauss point level homogenized first P-K stress is used to compute internal forces at macroscale finite element nodes

If these internal forces are in balance with the prescribed macro load, incremental convergence has been achieved and steps $1-6$ are repeated. If that is not the case, steps $2-6$ are repeated. The algorithmic flowchart is given in Algorithm 1.

Algorithm 1 Machine learning based FE^2 homogenization for elastic solids

```
Use machine learning to establish the relationship (5)
F_M ← I                                          ▷ Initialize deformation gradient
for E ∈ T_h do                                   ▷ loop over macroscale finite elements
    for g ∈ G do                                 ▷ loop over gauss points
        RVE ↔ g                                  ▷ Assign a RVE to each gauss point
        Compute macroscopic tangent stiffness at each gauss point in accordance with (13)
    Assemble macroscopic tangent stiffness over gauss points
Assemble macroscopic tangent stiffness over finite elements
while t ≤ T do
    Apply increment of macro load
    while (Internal force-Macro load > TOL) do   ▷ at macroscale finite element nodes
        Solve macroscale problem for δF_M
        F_M ← F_M + δF_M                         ▷ Update deformation gradient
        for E ∈ T_h do                           ▷ loop over macroscale finite elements
            for g ∈ G do                         ▷ loop over gauss points
                Prescribe periodic BCs in accordance with (6)
                Calculate first P-K stress in accordance with (5), (4) and (1)
        Compute internal forces at finite element nodes
```

Conflict of interest

The authors declare that they have no conflict of interest.

A The deformation gradient and first P-K stress

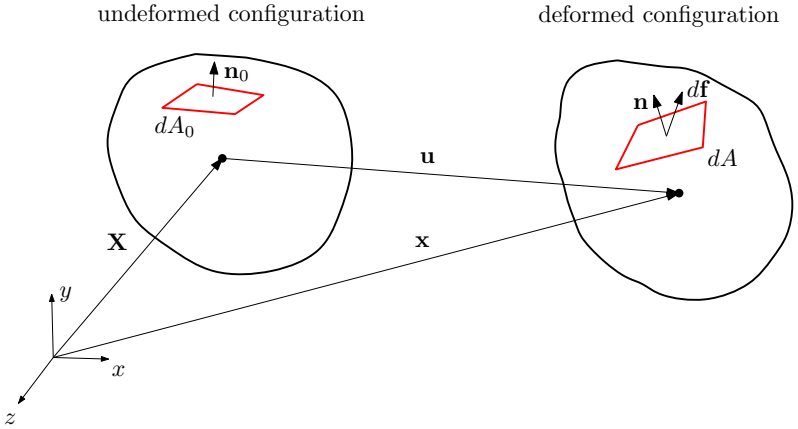

Fig. 4 $\mathbf{X}$ is position vector of point in reference configuration and $\mathbf{x} = \mathbf{X} + \mathbf{u}$ is the position vector the same point in the deformed configuration. Meanwhile, an elemental area dA_0 with unit normal $\mathbf{n}_0$ deforms to dA with unit normal $\mathbf{n}$ under the transformation.

As shown in Figure 4, let $\mathbf{u}$ be the macroscale deformation field. The macroscale deformation gradient $\mathbf{F}_M$ is the macroscale spatial derivative of $\mathbf{x}$ in the reference configuration as follows

$$\mathbf{F} = \mathbf{x} \otimes \nabla_X \equiv \mathbf{I} + \mathbf{u} \otimes \nabla_X$$

An incremental force $d\mathbf{f}$ is defined with respect to the Cauchy stress $\boldsymbol{\sigma}$ and the first Piola-Kirchoff stress $\mathbf{P}$ in the deformed and reference configurations respectively as follows

$$d\mathbf{f} = \boldsymbol{\sigma}\mathbf{n}\, dA = \mathbf{P}\mathbf{n}_0\, dA_0$$

B Periodic boundary conditions on RVE

The typical RVE for imposition of periodic boundary conditions is shown in Figure 5. After each macroscale BVP solve, the deformation gradient is updated and the new position vectors of the vertices of the RVE are obtained using

$$\mathbf{x} = \mathbf{F}_M \mathbf{X} \tag{6}$$

where $\mathbf{X}$ represents position vector in the reference configuration. This alongwith the shape periodicity of the RVE enables the implementation of periodic boundary conditions on RVE. It is easy to see that the prescribed periodic boundary conditions are Dirichlet boundary conditions.

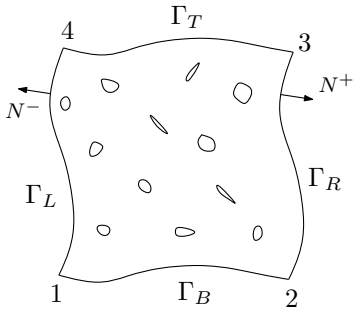

Fig. 5 Typical 2D RVE with pertinent microstructural features. Γ_L and Γ_R are mirror images so are Γ_T and Γ_B. This helps in easy implementation of periodic boundary conditions on the RVE in accordance with [12].

C Neural network

A neural network is composed of several connected layers of artificial neurons and biases where the data is fed into the input layer and flows through some hidden layers. The output is predicted in the output layer. The neurons from different layers are connected through weights w. In the data collection phase, the data flows in one way from the input layer to the target. At each neuron, an activation function is attached. The output of each neuron is computed by multiplying the outputs from the previous layer with the corresponding weights. For the neuron j in layer k, the data from the previous layer $k-1$ is summed up and then altered by an activation function. The output of neuron j in layer k is computed as

$$o_j^k = \mathcal{F}(\sum_{i=1}^{N} w_{ij} o_i^{k-1} + b_i^{k-1})$$

where N is the number of neurons in the previous layer $k-1$, w_{ij} is the weight connecting neurons i and j, o_i^{k-1} is the output of neuron i in layer $k-1$ and b_i^{k-1} is its bias. A common choice for the activation function is the sigmoid

$$\mathcal{F} = \frac{2}{1+e^{-2x}} - 1$$

In the training phase, the weights of neural network will be initialized firstly, (see [13]), which is followed by the weights updating using a training algorithm such that the global error is minimized. The global error, also named as loss function or network performance, is defined according to the difference between the network prediction and the target data. To minimize the global error, the Levenberg-Marquardt algorithm (see [14]) is applied to update the weights.

D Computation of homogenized first P-K stress at the macroscale

The linear momentum balance for the macroscale BVP in the reference configuration is given by

$$\nabla_X \cdot \mathbf{P}_M + \mathbf{b} = \mathbf{0} \tag{7}$$

where $\mathbf{b}$ is the body force vector. Equation (7) is expressed in the indicial notation as

$$P_{ik,k} + b_i = 0 \qquad i,k = 1,2,3$$

where the notation $(\cdot)_{,k} \equiv \frac{\partial(\cdot)}{\partial X_k}$ is used to denote the spatial derivative in the reference configuration. Before we proceed, we assume that the body force is zero, and obtain the

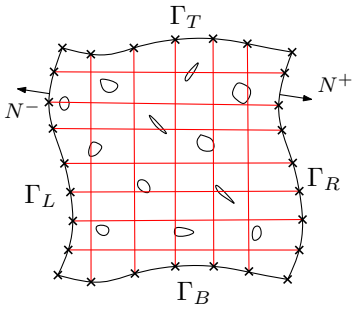

Fig. 6 Typical 2D RVE with finite element mesh. Boundary nodes are marked with crosses.

following using chain rule for differentiation

$$(P_{ik}X_j)_{,k} = P_{ik,k}X_j + P_{ik}\delta_{jk} = \cancelto{0}{-b_i}X_j + P_{ij} \tag{8}$$

We express the macroscale first P-K stress in indicial notation as follows

$$P_{M_{ij}} = \frac{1}{V_0}\int_{V_0} P_{ij}\, dV_0 = \frac{1}{V_0}\int_{V_0} (P_{ik}X_j)_{,k}\, dV_0 = \frac{1}{V_0}\int_{\Gamma_0} P_{ik}n_{0_k}X_j\, d\Gamma_0 \tag{9}$$

where the third equality follows from (8) and the fourth equality follows from divergence theorem. We then write (9) in tensorial notation as

$$\mathbf{P}_M = \frac{1}{V_0}\int_{\Gamma_0} \mathbf{t}_0 \otimes \mathbf{X}\, d\Gamma_0 \tag{10}$$

E Computation of homogenized tangent stiffness at macroscale

Let $\mathbf{u}_f$ represent the displacement DOFs corresponding to the interior nodes and $\mathbf{u}$ represent the displacement DOFs corresponding to the boundary nodes. The force displacement relation for the RVE scale problem is

$$\cancelto{\mathbf{K}^{RVE}}{\begin{bmatrix}\mathbf{K}_{pp} & \mathbf{K}_{pf}\\ \mathbf{K}_{fp} & \mathbf{K}_{ff}\end{bmatrix}}\begin{Bmatrix}\delta\mathbf{u}_p\\ \delta\mathbf{u}\end{Bmatrix} = \begin{Bmatrix}\delta\mathbf{f}\\ \mathbf{0}\end{Bmatrix}$$

where the matrix $\mathbf{K}^{RVE}$ is dictated by the microstructure and is known apriori. We knock off DOFs corresponding to internal nodes to obtain

$$\cancelto{\mathbf{K}}{[\mathbf{K}_{pp} - \mathbf{K}_{pf}(\mathbf{K}_{ff})^{-1}\mathbf{K}_{fp}]}\{\delta\mathbf{u}\} = \{\delta\mathbf{f}\} \tag{11}$$

The incremental macroscopic first PK stress is obtained as

$$\begin{aligned}\delta\mathbf{P}_M &= \frac{1}{V_0}\sum_{i=1}^{N}\delta\mathbf{f}_i \otimes \mathbf{X}_i \qquad (from\,(1))\\ &= \frac{1}{V_0}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{ij}\delta\mathbf{u}_j \otimes \mathbf{X}_i \qquad (from\,(11))\\ &= \frac{1}{V_0}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{ij}\delta\mathbf{F}_M\mathbf{X}_j \otimes \mathbf{X}_i \qquad (\delta\mathbf{u} = \delta\mathbf{F}_M\mathbf{X})\end{aligned} \tag{12}$$

Comparing (12) with $\delta\mathbf{P}_M = \mathbb{C}_M \delta\mathbf{F}_M$, we get

$$\mathbb{C}_{M_{abcd}} = \frac{1}{V_0} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{K}_{ij_{ac}} \mathbf{X}_{i_b} \mathbf{X}_{i_d} \qquad a, b, c, d = 1, 2, 3 \tag{13}$$

References

1. Z. Hashin and S. Shtrikman. On some variational principles in anisotropic and nonhomogeneous elasticity. *Journal of the Mechanics and Physics of Solids*, 10(4):335–342, 1962.
2. R. Hill. Elastic properties of reinforced solids: Some theoretical principles. *Journal of the Mechanics and Physics of Solids*, 11(5):357–372, 1963.
3. Saumik Dana. A simple framework for arriving at bounds on effective moduli in heterogeneous anisotropic poroelastic solids. *arXiv preprint arXiv:1912.10835*, 2019.
4. Saumik Dana, Joel Ita, and Mary F Wheeler. The correspondence between voigt and reuss bounds and the decoupling constraint in a two-grid staggered algorithm for consolidation in heterogeneous porous media. *Multiscale Modeling & Simulation*, 18(1):221–239, 2020.
5. Saumik Dana and Mary F Wheeler. An efficient algorithm for numerical homogenization of fluid filled porous solids: part-i. *arXiv preprint arXiv:2002.03770*, 2020.
6. Saumik Dana and Mary F Wheeler. A machine learning accelerated fe^2 homogenization algorithm for elastic solids. *arXiv preprint arXiv:2003.11372*, 2020.
7. M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175 – 2182, 2010. Fourth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2008).
8. I. Ozdemir, W. A. M. Brekelmans, and M. G. D. Geers. Fe2 computational homogenization for the thermo-mechanical analysis of heterogeneous solids. *Computer Methods in Applied Mechanics and Engineering*, 198(3):602 – 613, 2008.
9. Jörg Schröder. A numerical two-scale homogenization scheme: the fe2-method. In *Plasticity and beyond*, pages 1–64. Springer, 2014.
10. Dengpeng Huang, Jan Niklas Fuhg, Christian Weißenfels, and Peter Wriggers. A machine learning based plasticity model using proper orthogonal decomposition. *arXiv preprint arXiv:2001.03438*, 2020.
11. Manyu Xiao, Piotr Breitkopf, Rajan Filomeno Coelho, Catherine Knopf-Lenoir, Maryan Sidorkiewicz, and Pierre Villon. Model reduction by cpod and kriging. *Structural and multidisciplinary optimization*, 41(4):555–574, 2010.
12. V. V. Kouznetsova, W. A. M. Brekelmans, and F. P. T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. *Computational Mechanics*, 27(1):37–48, 2001.
13. Derrick Nguyen and Bernard Widrow. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *1990 IJCNN International Joint Conference on Neural Networks*, pages 21–26. IEEE, 1990.
14. Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.