Comparative study of three techniques for the computation of the macroscopic tangent moduli by periodic homogenization scheme

J. C. Zhu[1] · M. Ben Bettaieb[1,2] · F. Abed-Meraim[1,2]

Abstract

In numerical strategies developed for determining the effective macroscopic properties of heterogeneous media, the efficient and robust computation of macroscopic tangent moduli represents an essential step to achieve. Indeed, these tangent moduli are usually required in several numerical applications, such as the FE^2 method and the prediction of the onset of material and structural instabilities in heterogeneous media by loss of ellipticity approaches. This paper presents a comparative study of three numerical techniques for the computation of such tangent moduli in the context of periodic homogenization: the perturbation technique, the condensation technique and the fluctuation technique. The practical implementations of these techniques within ABAQUS/Standard finite element (FE) code are especially underlined. These implementations are based on the development of a set of Python scripts, which are connected to the finite element computations to handle the computation of the tangent moduli. The extension of these techniques to mechanical problems exhibiting symmetry properties is also detailed in this contribution. The reliability, accuracy and ease of implementation of these techniques are evaluated through some typical numerical examples. It is shown from this numerical and technical study that the condensation method reveals to be the most reliable and efficient. Also, this paper provides valuable reference guidelines to ABAQUS/Standard users for the determination of the homogenized tangent moduli of linear or nonlinear heterogeneous materials, such as composites, polycrystalline aggregates and porous solids.

Keywords Heterogeneous media · Finite strain · Periodic homogenization · Unit cell · Homogenized tangent moduli

1 Introduction

The main objective of micromechanical multiscale approaches is to determine the effective (also called homogenized, overall or macroscopic) mechanical properties of heterogeneous media under some specific boundary conditions. In this context, Hashin and Shtrikman [1] and Hill [2] have analytically estimated the overall properties of reinforced composites made of linear elastic phases. As various nonlinear composites (composites containing at least one nonlinear phase) have attracted special interest for both academic and industrial communities, the earlier pioneering works have subsequently been extended to determine their effective properties. For instance, Ponte Castaneda and Willis [3] have studied the mechanical behavior of nonlinear viscous composites. Suquet [4] has derived several analytical averaging relations for perfectly plastic composites. Teply and Dvorak [5] have investigated the overall behavior of elastoplastic composites. A comprehensive review of analytical approaches, developed to estimate homogenized properties of heterogeneous composite materials, has been provided in several contributions (see, e.g., [6–10]). Despite their wide use, analytical approaches are unable to give accurate effective properties for complex microstructures (random morphology and spatial distribution of constitutive phases…) exhibiting strong geometric and material nonlinearities. To overcome these limitations, some numerical approaches have recently been developed, as alternative to analytical ones. In this regard, one may quote at least two

✉ M. Ben Bettaieb
  Mohamed.BenBettaieb@ensam.eu

1 CNRS, Arts et Metiers Institute of Technology, LEM3, Université de Lorraine, 57070 Metz, France

2 DAMAS, Laboratory of Excellence on Design of Alloy Metals for Low-mAss Structures, Université de Lorraine, Metz, France

types of commonly used approaches: those based on fast Fourier transforms (FFT), and those based on the finite element method (FEM). As to the FFT methods, Moulinec and his coauthors have proposed in [11–14] a variety of multiscale schemes and compared their convergence rates for computing overall properties of both linear and nonlinear composites. From these investigations, it has been shown that the contrast between the mechanical properties in the phases significantly affects the convergence rate of the FFT-based computations. Michel et al. [15] have demonstrated that FEM-based approaches converge more easily for composites with periodic microstructure and made of phases with infinite stiffness contrast (such as rigid inclusions or voids). Consequently, FEM-based approaches received more attention when dealing with more general heterogeneous materials, such as composites, polycrystalline aggregates and porous media. In this perspective, Miehe [16] has developed a strain-driven homogenization approach for inelastic microstructures and composites based on an incremental variational formulation. This formulation can be used to compute the macroscopic response of elastoviscoplastic media under the three well-known boundary conditions: (i) linear deformation (LD), (ii) uniform traction (UT), and (iii) periodic boundary conditions (PBC). A small strain formulation has been employed in [16] to establish the multiscale relations as well as the constitutive equations of the different phases. The multiscale schemes presented in [16] have been extended in Miehe [17] to finite strain problems, where the Lagrange multiplier method has been employed to enforce the above boundary conditions on the microstructure. This extension has been achieved by generating a family of algorithms that allow deriving homogenized stress and tangent moduli for composites and polycrystalline aggregates. In the present contribution, attention is focused on the modeling of the mechanical behavior of heterogeneous media exhibiting a periodic or quasi-periodic distribution of heterogeneities (such as composite materials, voided media, or polycrystalline aggregates). Considering this spatial periodicity, the periodic homogenization approach turns out to be the more suitable scheme to ensure the transition between microscopic and macroscopic levels. The studied media are assumed to undergo large deformation. Consequently, a total Lagrangian formulation is adopted to formulate the governing equations of the periodic homogenization approach, where the deformation gradient and the first Piola–Kirchhoff stress tensor are selected as appropriate strain and stress measures, respectively. As first step for the application of the periodic homogenization approach, a unit cell assumed to be representative of the heterogeneous medium should be selected. Several studies, mainly based on some statistical techniques, have been carried out in the literature to evaluate the minimum size of the unit cell to ensure its representativeness of the heterogeneous medium.

This step is not the object of the current contribution and the interested readers may refer to reference [18]. Once the unit cell is defined, the equations governing the periodic homogenization problem (namely, localization and homogenization relations, microscopic equilibrium equations, periodic boundary conditions) are solved by the finite element method. To this end, several academic finite element codes, such as Lagamine [19] or Zébulon [20], have been extended by implementing some built-in tools and routines to automatically achieve the application of the macroscopic loading and periodic boundary conditions and the computation of the macroscopic response. Analogous to most popular commercial finite element codes, such as ABAQUS/ Standard, several plugin tools have been developed to easily apply the periodic boundary conditions as well as the macroscopic loading (which can be strain-driven or stress-driven) and to compute the macroscopic response. Among these tools, one may quote the toolbox *Homtools* developed by Lejeunes and Bourgeois [21], the *EasyPBC* tool proposed in [22] or the Python codes developed in [23]. Unfortunately, the above-mentioned tools are unable to achieve the automatic computation of the macroscopic tangent moduli. On the other hand, particular attention has been paid, in some contributions, to the efficient and accurate computation of these moduli for some engineering numerical applications using multiscale schemes. Indeed, the macroscopic tangent modulus is required for the prediction of the mechanical behavior of polycrystalline structures by the FE[2] method [24–29]. It is also needed for the prediction of the onset of macroscopic material and structural instabilities by loss of ellipticity approaches [30–33]. In the literature, at least three FEM-based techniques have been developed to determine the macroscopic tangent modulus from the periodic homogenization computations:

- The perturbation technique (PT): this technique allows reducing the computation of the macroscopic tangent modulus to multiple macroscopic stress computations [23, 34]. By perturbing the macroscopic deformation gradient, the macroscopic tangent modulus is efficiently constructed by a forward difference of the macroscopic first Piola–Kirchhoff stress. To obtain the approximation of the ith column of the macroscopic tangent modulus, a small perturbation of the ith component of the macroscopic deformation gradient is needed (for 3D finite strain, where i ranges between 1 and 9). The ith column of the macroscopic tangent modulus is equal to the forward difference between the perturbed and unperturbed macroscopic stress divided by the corresponding perturbation of the macroscopic deformation gradient. This method, widely adopted to numerically evaluate the tangent modulus for several phenomenological models [35], has recently been coupled with the periodic homog-

enization scheme in some investigations [23, 34]. This technique has the merit of being conceptually easy to understand, but generally requires a great deal of CPU time, as the finite element computation should be performed ten times for each increment (one time to compute the macroscopic unperturbed stress and nine times to construct the macroscopic tangent modulus).

- The condensation technique (CT): within this technique, the macroscopic tangent modulus is obtained by a condensation procedure of the global stiffness matrix. This method has initially been introduced for a small strain formulation [36], and subsequently extended to finite strain framework [17]. The practical application of this technique is based on the construction of a family of link-topology matrices to automatically represent the overall properties of discretized microstructures. The updated global stiffness matrix as well as the initial coordinates of the nodes on the boundary of the unit cell are required to compute the macroscopic tangent modulus by the condensation method.

- The fluctuation technique (FT): in this technique, the deformation gradient of each node of the FE discretization is additively decomposed into two parts: a homogeneous part associated with the macroscopic loading, and a nonhomogeneous part resulting from the periodic fluctuation contribution. As a result of this decomposition, the macroscopic tangent modulus can be written as the sum of the volume average of tangent moduli over the unit cell, and a fluctuation part, which depends on the global stiffness matrix and on a fluctuation matrix. The volume average of microscopic tangent moduli yields the well-known Taylor-type upper bound. The numerical procedures of this method at small strain and finite strain have been presented in [37, 38], respectively.

As ABAQUS/Standard is used to perform the finite element computations, the above-presented three techniques have been implemented in a set of Python scripts. The choice of the Python language is motivated by the fact that ABAQUS/Standard has a built-in Python API (application programming interface) that offers a number of benefits in extracting data from an ABAQUS database. Furthermore, some popular numerical packages, such as the NumPy and SciPy libraries, are used in developing the Python scripts. These libraries contain a powerful *N*-dimensional array object and useful linear algebra methods, thus enabling efficient matrix computations (reduced computation time and memory space). These python scripts are interpreted as post-processing of the finite element analysis. The toolbox *Homtools* [21] is used to automatically determine and generate the required boundary node sets, constraint equations, periodic boundary conditions, and post-processing calculations in order to compute the macroscopic response. Technical

details, related to the connection between the ABAQUS/ Standard FE code environment and the developed Python codes, will be provided in this investigation. All of the developed techniques can be applied for any loading path and at each finite element increment. The CPU time required to perform these analyses is mainly dependent on the number of time increments, at which the macroscopic tangent modulus is computed, and on the number of the degrees of freedom (DOFs) of the finite element mesh. These analyses could induce a very high computation cost (computation time and memory space) if we intend to compute the macroscopic tangent modulus at each finite element increment. To reduce the computation cost involved in such analyses, without losing the accuracy of the finite element solution, we have offered the possibility of computing the macroscopic tangent modulus for each *n* finite element increments (computation at the end of the *n*th, 2*n*th, 3*n*th finite element increment…). A comparative study between the above-discussed three techniques will be given in the section corresponding to the numerical results. Thus, this paper gives valuable reference guidelines to ABAQUS/Standard users for the determination of the homogenized tangent moduli. On the basis of this study, it is demonstrated that the condensation technique reveals to be the most efficient method (easy to implement, requires less CPU time and disk space). The computation of the tangent moduli corresponding to microstructures exhibiting symmetry properties is also detailed in the current contribution. Note that such extension to symmetric problems allows considerably improving the computational performance.

The remainder of the paper is organized as follows:

- Sect. 2 provides some technical details about the stress rates and associated tangent moduli adopted in ABAQUS/Standard FE code (built-in model or user-defined model) to formulate and solve the weak form of the virtual work principle. These details are essential for the understanding of the subsequent sections.
- The formulation of the periodic homogenization problem at finite strain, and the practical aspects related to the solution of this problem are discussed in Sect. 3.
- The numerical aspects and the operational details related to the implementation of the three techniques for the computation of the tangent modulus (namely, the perturbation technique, the condensation technique, and the fluctuation technique) are detailed in Sect. 4.
- Sect. 5 is dedicated to the adaptation of the perturbation and condensation techniques to the case of microstructures displaying symmetry properties.
- In Sect. 6, the implementation of the three techniques is validated by comparing their results with those published in [37]. The performances of these techniques are

reported, discussed and compared through some numerical examples.

1.1 Conventions, notations and abbreviations

The following conventions and notations are used throughout:

- Microscale (resp. macroscale) variables are denoted by lowercase (resp. capital) letters.
- Vectors and tensors are indicated by bold letters or symbols. However, scalar parameters and variables are designated by thin and *italic* letters or symbols.
- $\dot{\bullet}$ time derivative of $\bullet$.
- $\bullet^T$ transpose of $\bullet$.
- $\bullet^{-1}$ inverse of $\bullet$.
- $\det(\bullet)$ determinant of $\bullet$.
- $^v\bullet$ vector representation of the second-order tensor $\bullet$. ($^v\bullet := \begin{bmatrix} \bullet_{11} & \bullet_{22} & \bullet_{33} & \bullet_{12} & \bullet_{23} & \bullet_{13} & \bullet_{21} & \bullet_{32} & \bullet_{31} \end{bmatrix}^T$).
- $^m\bullet$ matrix representation of the fourth-order tensor $\bullet$ (defined similarly to the vector representation).
- $\bullet \cdot \bullet$ simple contraction or contraction on one index (inner product).
- $\bullet : \bullet$ double contraction or contraction on two indices (inner product).
- $\bullet \otimes \bullet$ tensor product (external product).
- δ_{ij} Kronecker delta.
- $\mathbf{I}_2$ second-order identity tensor.
- $\mathcal{D}$ iteration of field $\bullet$ (within an incremental finite element computation).
- $\Delta\bullet$ small perturbation used for the differentiation of field $\bullet$.
- PT   perturbation technique.
- CT   condensation technique.
- FT   fluctuation technique.
- FUC   full unit cell (without or before application of symmetry restrictions).
- RUC   reduced unit cell (after application of symmetry restrictions).
- UEL   user element subroutine.
- UMAT   user material subroutine.
- DOFs   degrees of freedom.

2 Details on the finite element formulation in ABAQUS/Standard

The majority of commercial finite element codes (such as ABAQUS, ANSYS, NASTRAN, LS-DYNA…) allow the accurate computation of the mechanical response of solids and structures exhibiting strong material and geometric non-linearities (elastoplastic behavior, finite strain, finite rotation, contact…). Within the finite strain framework, there are several work-conjugate strain/stress measures classically used to formulate the virtual work principle, which is the basis of the finite element method. For instance, the ABAQUS/Standard built-in formulation[1] is based on the Jaumann rate of the Kirchhoff stress tensor, while NASTRAN is based on the Truesdell rate of the Cauchy stress tensor. The finite element simulation results must be independent of the choice of these strain/stress measures and the associated virtual work form. Hence, to ensure the accuracy and correctness of the finite element results, the specific stress rate and associated tangent modulus (relating the stress rate to the associated strain rate) should be properly chosen. The current section provides a brief overview of the relations between the different stress rates as well as the associated tangent moduli commonly used in ABAQUS/Standard code.

Starting with the formulation of the virtual work principle, the equilibrium equation is expressed in a rate form as follows:

$$\text{div}_{\mathbf{x}_0}(\dot{\mathbf{p}}) + \dot{\mathbf{b}}_0 = \mathbf{0}, \tag{1}$$

where: $\dot{\mathbf{p}}$ is the rate of the first Piola–Kirchhoff stress and $\text{div}_{\mathbf{x}_0}(\dot{\mathbf{p}})$ its divergence with respect to the reference coordinate system $\mathbf{x}_0$. $\dot{\mathbf{b}}_0$ is the body force rate per unit volume in the reference configuration.

Multiplying Eq. (1) by a virtual velocity field $\delta\mathbf{v}$ and integrating over the volume of the reference configuration $\mathcal{V}_0$ yields:

$$\int_{\mathcal{V}_0} \delta\mathbf{v} \cdot \left[\text{div}_{\mathbf{x}_0}(\dot{\mathbf{p}}) + \dot{\mathbf{b}}_0\right] d\mathcal{V}_0 = 0. \tag{2}$$

Employing the chain rule and Gauss theorem, Eq. (2) can be reformulated as follows:

$$\int_{\mathcal{V}_0} \left[\frac{\partial \delta\mathbf{v}}{\partial \mathbf{x}_0} : \dot{\mathbf{p}}\right] d\mathcal{V}_0 = \int_{\mathcal{S}_0} \delta\mathbf{v} \cdot \dot{\mathbf{t}}_0 d\mathcal{S}_0 + \int_{\mathcal{V}_0} \delta\mathbf{v} \cdot \dot{\mathbf{b}}_0 d\mathcal{V}_0, \tag{3}$$

where $\mathcal{S}_0$ and $\dot{\mathbf{t}}_0$ denote the boundary surface of the reference configuration and the nominal traction rate prescribed on $\mathcal{S}_0$, respectively. Vector $\dot{\mathbf{t}}_0$ is equal to $\dot{\mathbf{p}} \cdot \vec{\mathbf{n}}_0$, where $\vec{\mathbf{n}}_0$ is the outer normal to the boundary $\mathcal{S}_0$.

The virtual work principle defined by Eq. (3) can be equivalently expressed in terms of the Kirchhoff stress $\boldsymbol{\tau}$ ($= \mathbf{p} \cdot \mathbf{f}^T$, where $\mathbf{f}$ denotes the deformation gradient) and its objective derivatives $\tilde{\boldsymbol{\tau}}^{(m)}$ as follows [39]:

---

[1] It is referred to ABAQUS/Standard formulation without user subroutines for nonlinear incremental analysis in this paper. ABAQUS/Explicit adopts different fundamental solving technique, as well as the theoretical formulation.

$$\int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \tilde{\boldsymbol{\tau}}^{(m)} - (2-m)\boldsymbol{\tau} : (\delta\mathbf{d} \cdot \mathbf{d}) + \boldsymbol{\tau} : (\delta\mathbf{l}^T \cdot \mathbf{l}) \right] d\mathcal{V}_0$$
$$= \int_{\mathcal{S}_0} \delta\mathbf{v} \cdot \dot{\mathbf{t}}_0 d\mathcal{S}_0 + \int_{\mathcal{V}_0} \delta\mathbf{v} \cdot \dot{\mathbf{b}}_0 d\mathcal{V}_0, \tag{4}$$

where $\mathbf{l}$ is the velocity gradient, defined as $\partial\mathbf{v}/\partial\mathbf{x}$, and $\delta\mathbf{l}$ its virtual counterpart. $\mathbf{d}$ is the strain rate, defined as the symmetric part of $\mathbf{l}$, and $\delta\mathbf{d}$ its virtual counterpart. m is a parameter that defines the different objective rates of the Kirchhoff stress. It is equal to 0, 1, and 2 for the Jaumann, Biot, and Truesdell rates, respectively.

In ABAQUS/Standard built-in modeling, the Jaumann rate is employed. Hence, Eq. (4) is used with $m = 0$:

$$\int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \tilde{\boldsymbol{\tau}}^{(0)} - 2\boldsymbol{\tau} : (\mathbf{d} \cdot \delta\mathbf{d}) + \boldsymbol{\tau} : (\mathbf{l}^T \cdot \delta\mathbf{l}) \right] d\mathcal{V}_0$$
$$= \int_{\mathcal{S}_0} \delta\mathbf{v} \cdot \dot{\mathbf{t}}_0 d\mathcal{S}_0 + \int_{\mathcal{V}_0} \delta\mathbf{v} \cdot \dot{\mathbf{b}}_0 d\mathcal{V}_0. \tag{5}$$

On the other hand, the Jaumann rate of the Kirchhoff stress $\tilde{\boldsymbol{\tau}}^{(0)}$ is related to the strain rate $\mathbf{d}$ as follows:

$$\tilde{\boldsymbol{\tau}}^{(0)} = \mathbf{c}^{(0)} : \mathbf{d}, \tag{6}$$

where the tangent modulus $\mathbf{c}^{(0)}$ is expressed in terms of the Jacobian matrix **DDSDDE** (using the terminology of ABAQUS/Standard FE code) and $j (= \det(\mathbf{f}))$ as follows:

$$\mathbf{c}^{(0)} = j\,\mathbf{DDSDDE}. \tag{7}$$

Utilizing finite element discretization, the velocity field $\mathbf{v}$ and the strain rate $\mathbf{d}$ can be obtained from the nodal displacement rate vector $\dot{\mathbf{q}}$ by the interpolation rule:

$$\mathbf{v} = \mathbf{N} \cdot \dot{\mathbf{q}}; \quad \mathbf{d} = \frac{1}{2}\left(\mathbf{B} + \mathbf{B}^T\right) \cdot \dot{\mathbf{q}}, \tag{8}$$

with $\mathbf{N}$ being the shape function vector, and $\mathbf{B}$ its derivative with respect to $\mathbf{x}$.

The substitution of Eqs. (6) and (8) into the different components of Eq. (5) leads to:

$$\begin{cases} \int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \tilde{\boldsymbol{\tau}}^{(0)}\right] d\mathcal{V}_0 = \int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \mathbf{c}^{(0)} : \mathbf{d}\right] d\mathcal{V}_0 = \delta\dot{\mathbf{q}}^T \cdot \left[\int_{\mathcal{V}_0} \mathbf{B}^T \cdot {}^m\mathbf{c}^{(0)} \cdot \mathbf{B} d\mathcal{V}_0\right] \cdot \dot{\mathbf{q}} = \delta\dot{\mathbf{q}}^T \cdot \mathbf{K}_M \cdot \dot{\mathbf{q}}; \\ \int_{\mathcal{V}_0} \left[-2\boldsymbol{\tau} : (\mathbf{d} \cdot \delta\mathbf{d}) + \boldsymbol{\tau} : (\mathbf{l}^T \cdot \delta\mathbf{l})\right] d\mathcal{V}_0 = \delta\dot{\mathbf{q}}^T \cdot \mathbf{K}_{GNL} \cdot \dot{\mathbf{q}}; \\ \int_{\mathcal{S}_0} \dot{\mathbf{t}}_0 \cdot \delta\mathbf{v} d\mathcal{S}_0 + \int_{\mathcal{V}_0} \dot{\mathbf{b}}_0 \cdot \delta\mathbf{v} d\mathcal{V}_0 = \delta\dot{\mathbf{q}}^T \cdot \dot{\mathbf{r}}, \end{cases} \tag{9}$$

where $\mathbf{K}_M$ is the part of the global stiffness matrix associated with the mechanical behavior. $\mathbf{K}_{GNL}$ is the part of the global stiffness matrix resulting from the geometric nonlinearities. $\dot{\mathbf{r}}$ is the residual vector.

Then, Eq. (5) can be written in the following discretized matrix form:

$$\left[\mathbf{K}_M + \mathbf{K}_{GNL}\right] \cdot \dot{\mathbf{q}} = \mathbf{K} \cdot \dot{\mathbf{q}} = \dot{\mathbf{r}}. \tag{10}$$

Equation (10) is strongly nonlinear, as stiffness matrix $\mathbf{K}$ $(= \left[\mathbf{K}_M + \mathbf{K}_{GNL}\right])$ and vector $\dot{\mathbf{r}}$ depend on the nodal displacement rate vector $\dot{\mathbf{q}}$. Consequently, this equation is classically solved by an iterative scheme, such as the Newton–Raphson method.

For ABAQUS/Standard FE code, when the mechanical behavior is inelastic (elastoplastic, elastoviscoplastic…) and built-in material subroutines are used to model this behavior (i.e., without user material subroutine UMAT), Eq. $(9)_1$ is rearranged as follows:

$$\int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \tilde{\boldsymbol{\tau}}^{(0)}\right] d\mathcal{V}_0 = \int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \mathbf{c}^e : \mathbf{d}^e\right] d\mathcal{V}_0$$
$$= \int_{\mathcal{V}_0} \left[\delta\mathbf{d} : \mathbf{c}^e : (\mathbf{d} - \mathbf{d}^p)\right] d\mathcal{V}_0$$
$$= \delta\dot{\mathbf{q}}^T \cdot \left(\left[\int_{\mathcal{V}_0} \left[\mathbf{B}^T \cdot {}^m\mathbf{c}^e \cdot \mathbf{B}\right] d\mathcal{V}_0\right] \cdot \dot{\mathbf{q}} - \int_{\mathcal{V}_0} \left[\mathbf{B}^T \cdot {}^m\mathbf{c}^e \cdot {}^v\mathbf{d}^p\right] d\mathcal{V}_0\right)$$
$$= \delta\dot{\mathbf{q}}^T \cdot \left(\mathbf{K}_E \cdot \dot{\mathbf{q}} - \dot{\mathbf{r}}_p\right), \tag{11}$$

where $\mathbf{d}^e$ and $\mathbf{d}^p$ are the elastic and plastic parts of the strain rate, and $\mathbf{c}^e$ is the elasticity modulus, which contributes to the elastic stiffness $\mathbf{K}_E$. With this rearrangement, the global nonlinear equation system (10) is reformulated as follows:

$$\left[\mathbf{K}_E + \mathbf{K}_{GNL}\right] \cdot \dot{\mathbf{q}} = \dot{\mathbf{r}} + \dot{\mathbf{r}}_p. \tag{12}$$

Hence, when the ABAQUS/Standard built-in mechanical modeling is used, the global stiffness matrix is only constructed on the basis of the elastic contribution. However, when user material subroutines UMAT are used, the stiffness matrix is constructed on the basis of the **DDSDDE** tangent modulus, which obviously considers the effect of both elastic and inelastic behavior.

On the other hand, the rate of the first Piola–Kirchhoff stress $\dot{\mathbf{p}}$ is related to the rate of the deformation gradient $\dot{\mathbf{f}}$ by the tangent modulus $\mathbf{c}^{(PK1)}$:

$$\dot{\mathbf{p}} = \mathbf{c}^{(PK1)} : \dot{\mathbf{f}}. \tag{13}$$

The substitution of Eq. (13) into Eq. (3) leads to the following expression of the virtual work principle:

$$\int_{\mathcal{V}_0} \left[\delta\dot{\mathbf{f}} : \mathbf{c}^{(PK1)} : \dot{\mathbf{f}}\right] d\mathcal{V}_0 = \int_{\mathcal{S}_0} \delta\mathbf{v} \cdot \dot{\mathbf{t}}_0 d\mathcal{S}_0 + \int_{\mathcal{V}_0} \delta\mathbf{v} \cdot \dot{\mathbf{b}}_0 d\mathcal{V}_0. \tag{14}$$

Equation (14) can be considered as the total Lagrangian formulation of the virtual work principle. Forms (5) and (14) are strictly equivalent as long as the relationship between $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(PK1)}$ is properly defined. To define this relationship, let us introduce tangent moduli $\mathbf{c}^{(2)}$ and $\mathbf{c}^{(PK2)}$ relating the Truesdell derivative of the Kirchhoff stress $\tilde{\boldsymbol{\tau}}^{(2)}$ ($m = 2$) to the strain rate $\mathbf{d}$, on the one hand, and the second Piola–Kirchhoff stress rate $\dot{\mathbf{s}}$ to the Green strain rate $\dot{\mathbf{e}}$, on the other hand:

$$\tilde{\boldsymbol{\tau}}^{(2)} = \mathbf{c}^{(2)} : \mathbf{d} \quad ; \quad \dot{\mathbf{s}} = \mathbf{c}^{(PK2)} : \dot{\mathbf{e}}. \tag{15}$$

As demonstrated in [39], $\mathbf{c}^{(0)}$ is linked to $\mathbf{c}^{(2)}$ through the following indicial form:

$$\forall i,j,k,l = 1,2,3 : \quad c_{ijkl}^{(0)} = c_{ijkl}^{(2)} + \frac{1}{2}\left(\tau_{ik}\delta_{jl} + \tau_{jk}\delta_{il} + \tau_{il}\delta_{jk} + \tau_{jl}\delta_{ik}\right), \tag{16}$$

where δ_{ij} is the Kronecker delta.

Meanwhile, $\mathbf{c}^{(2)}$ and $\mathbf{c}^{(PK1)}$ are related to $\mathbf{c}^{(PK2)}$ by the following indicial forms [39]:

$$\forall i,j,k,l = 1,2,3 : \quad c_{ijkl}^{(2)} = f_{im}f_{jn}f_{kp}f_{lq}c_{mnpq}^{(PK2)} \quad ; \quad c_{inkq}^{(PK1)} = c_{mnpq}^{(PK2)}f_{im}f_{kp} + s_{nq}\delta_{ik}. \tag{17}$$

Equation (16) together with Eq. (17) define the relations between the different tangent modulus forms (namely, $\mathbf{c}^{(0)}$, $\mathbf{c}^{(2)}$, $\mathbf{c}^{(PK1)}$ and $\mathbf{c}^{(PK2)}$). If these relations are not strictly respected, the two forms of the virtual work principle (i.e., Eqs. (5) and (14)) become different, and this difference may lead to some work-conjugacy issues (convergence problems, errors associated with lack of energy conservation…).

3 Periodic homogenization problem

The periodic homogenization technique is used to ensure the transition between microscopic and macroscopic scales. Without loss of generality, we assume that the initial configuration of the unit cell $\mathcal{V}_0$ occupies the domain $\left[-\lambda_0/2,\ \lambda_0/2\right] \times \left[-\lambda_0/2,\ \lambda_0/2\right] \times \left[-\lambda_0/2,\ \lambda_0/2\right]$. The studied unit cell is assumed to undergo finite strain and total Lagrangian formulation is used to formulate the periodic homogenization relations. Consequently, the deformation gradient and the first Piola–Kirchhoff stress tensor are used as appropriate work-conjugate strain and stress measures. For the sake of clarity, microscopic (resp. macroscopic) quantities will be denoted by small (resp. capital) characters and symbols.

The main equations governing the periodic homogenization scheme are outlined hereafter:

- The microscopic deformation gradient $\mathbf{f}$ is additively decomposed into its macroscopic counterpart $\mathbf{F}$ and a periodic fluctuation gradient $\mathbf{f}_{per}$:

$$\mathbf{f} = \mathbf{F} + \mathbf{f}_{per}. \tag{18}$$

The current position $\mathbf{x}$ of a material point can be determined by spatial integration of Eq. (18):

$$\mathbf{x} = \mathbf{F} \cdot \mathbf{x}_0 + \mathbf{u}_{per}, \tag{19}$$

where $\mathbf{x}_0$ is the initial position of the material point, and $\mathbf{u}_{per}$ is a periodic displacement over the initial configuration of the unit cell. The expression of the nodal velocity $\mathbf{v}$ can be easily derived from Eq. (19):

$$\mathbf{v} = \dot{\mathbf{F}} \cdot \mathbf{x}_0 + \dot{\mathbf{u}}_{per} := \dot{\mathbf{F}} \cdot \mathbf{x}_0 + \mathbf{v}_{per}. \tag{20}$$

- The averaging relations linking the microscopic deformation gradient $\mathbf{f}$ and the microscopic first Piola–Kirchhoff stress tensor $\mathbf{p}$ to their macroscopic counterparts $\mathbf{F}$ and $\mathbf{P}$:

$$\mathbf{F} = \frac{1}{|\mathcal{V}_0|}\int_{\mathcal{V}_0} \mathbf{f}\, d\mathcal{V}_0 \quad ; \quad \mathbf{P} = \frac{1}{|\mathcal{V}_0|}\int_{\mathcal{V}_0} \mathbf{p}\, d\mathcal{V}_0, \tag{21}$$

with $|\mathcal{V}_0|$ being the initial volume of the unit cell, which is equal here to $\left(\lambda_0\right)^3$.

Equation (21) can be equivalently expressed in the following rate form:

$$\dot{\mathbf{F}} = \frac{1}{|\mathcal{V}_0|}\int_{\mathcal{V}_0} \dot{\mathbf{f}}\, d\mathcal{V}_0 \quad ; \quad \dot{\mathbf{P}} = \frac{1}{|\mathcal{V}_0|}\int_{\mathcal{V}_0} \dot{\mathbf{p}}\, d\mathcal{V}_0, \tag{22}$$

- The microscopic static equilibrium equation in the absence of body forces:

$$\mathrm{div}_{\mathbf{x}_0}\left(\dot{\mathbf{p}}\right) = \mathbf{0}. \tag{23}$$

- The constitutive relation describing the microscopic mechanical behavior can be summarized by the following generic form:

$$\dot{\mathbf{p}} = \mathbf{c}^{(PK1)} : \dot{\mathbf{f}}. \tag{24}$$

Compared to classic finite element problems, the periodic homogenization problem defined by Eqs. (18)–(24) presents two particular specificities: the nature of the loading to which the unit cell is subjected, and the boundary conditions applied on the outer surfaces of the unit cell. These practical aspects will be clarified in the following developments. As a starting point for these developments, the virtual work

principle given in Eq. (3) is reduced to the following form in the absence of body forces:

$$\int_{\mathcal{V}_0} \left[\frac{\partial \delta \mathbf{v}}{\partial \mathbf{x}_0} : \dot{\mathbf{p}} \right] d\mathcal{V}_0 = \int_{\mathcal{S}_0} \delta \mathbf{v} \cdot \dot{\mathbf{t}}_0 d\mathcal{S}_0. \qquad (25)$$

Considering the decomposition in Eq. (20) and Eq. (22), the left-hand side of Eq. (25) can be written as follows:

$$\int_{\mathcal{V}_0} \left(\frac{\partial \delta \mathbf{v}}{\partial \mathbf{x}_0} : \dot{\mathbf{p}} \right) d\mathcal{V}_0 = \int_{\mathcal{V}_0} \left(\left(\delta \dot{\mathbf{F}} + \frac{\partial \delta \mathbf{v}_{per}}{\partial \mathbf{x}_0} \right) : \dot{\mathbf{p}} \right) d\mathcal{V}_0$$
$$= \int_{\mathcal{V}_0} \left(\delta \dot{\mathbf{F}} : \dot{\mathbf{p}} \right) d\mathcal{V}_0 + \int_{\mathcal{S}_0} \left(\left(\delta \mathbf{v}_{per} \otimes \vec{\mathbf{n}}_0 \right) : \dot{\mathbf{p}} \right) d\mathcal{S}_0. \qquad (26)$$

Considering the anti-periodicity of tensor $\delta \mathbf{v}_{per} \otimes \vec{\mathbf{n}}_0$ and the periodicity of $\dot{\mathbf{p}}^T$, one can easily show that:

$$\int_{\mathcal{S}_0} \left(\left(\delta \mathbf{v}_{per} \otimes \vec{\mathbf{n}}_0 \right) : \dot{\mathbf{p}} \right) d\mathcal{S}_0 = 0. \qquad (27)$$

Together with Eq. (27), Eq. (26) is reduced to the Hill–Mandel condition [40]:

$$\int_{\mathcal{V}_0} \left(\frac{\partial \delta \mathbf{v}}{\partial \mathbf{x}_0} : \dot{\mathbf{p}} \right) d\mathcal{V}_0 = |\mathcal{V}_0| \, \delta \dot{\mathbf{F}} : \dot{\mathbf{P}}. \qquad (28)$$

As shown in [34, 41, 42] for the classical periodic homogenization technique within a total Lagrangian formulation, Eq. (26) enables to treat the macroscopic deformation gradient rate $\dot{\mathbf{F}}$ as macroscopic DOFs associated with the nodal forces $|\mathcal{V}_0| \, \dot{\mathbf{P}}$. In practice, the application of macroscopic loading in combination with periodic boundary conditions within ABAQUS is ensured by the use of the reference point technique [21]. We detail in the following developments how the reference point technique manages the macroscopic loading and the periodic boundary conditions in only one space direction (direction 1). The extension to the other directions can be done in a similar way.

If we consider two nodes M^- and M^+ belonging to faces B_1^- and B_1^+ and having identical coordinates in 2 and 3 directions (Fig. 1), the position of these two nodes can be determined from Eq. (19):

$$\mathbf{x}_{M^-} = \mathbf{F} \cdot \mathbf{x}_{0M^-} + \mathbf{u}_{M^-}^{per} \quad ; \quad \mathbf{x}_{M^+} = \mathbf{F} \cdot \mathbf{x}_{0M^+} + \mathbf{u}_{M^+}^{per}. \qquad (29)$$

The periodic boundary conditions require that $\mathbf{u}_{M^-}^{per} = \mathbf{u}_{M^+}^{per}$. Consequently, Eq. (29)$_1$ and Eq. (29)$_2$ can be combined to obtain:

$$\mathbf{x}_{M^+} - \mathbf{x}_{M^-} = \mathbf{F} \cdot \left(\mathbf{x}_{0M^+} - \mathbf{x}_{0M^-} \right). \qquad (30)$$

On the other hand, the difference $\mathbf{x}_{M^+} - \mathbf{x}_{M^-}$ is defined as follows:
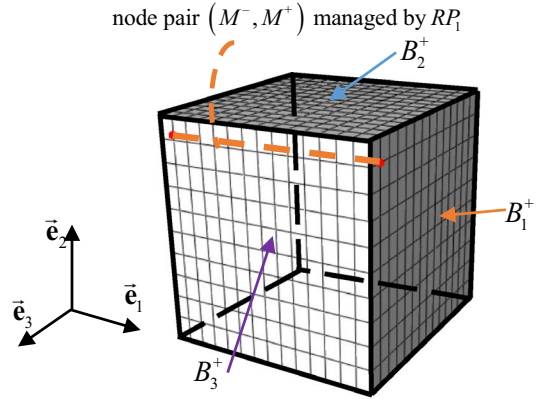

Fig. 1 Illustration of the periodic boundary conditions between nodes M^- and M^+

$$\mathbf{x}_{M^+} - \mathbf{x}_{M^-} = \left(\mathbf{u}_{M^+} - \mathbf{u}_{M^-} \right) + \left(\mathbf{x}_{0M^+} - \mathbf{x}_{0M^-} \right), \qquad (31)$$

where $\mathbf{u}_{M^+}$ and $\mathbf{u}_{M^-}$ are the displacements of nodes M^+ and M^-, respectively. Considering the initial coordinates of nodes M^+ and M^-, the substitution of Eq. (31) into Eq. (30) leads to the following relation between displacements $\mathbf{u}_{M^+}$ and $\mathbf{u}_{M^-}$:

$$\mathbf{u}_{M^+} - \mathbf{u}_{M^-} = \left(\mathbf{F} - \mathbf{I}_2 \right) \cdot \left(\mathbf{x}_{0M^+} - \mathbf{x}_{0M^-} \right) = \left(\mathbf{F} - \mathbf{I}_2 \right) \cdot \begin{bmatrix} \lambda_0 \\ 0 \\ 0 \end{bmatrix}. \qquad (32)$$

Practically, the periodic boundary conditions on the opposite faces B_1^- and B_1^+ and the macroscopic loading $\mathbf{F}$, summarized by Eq. (32), are applied by using the multipoint constraints (MPC) option of ABAQUS for each node pair $\left(M^-, M^+ \right)$. The set of python scripts *Homtools* enables to automatically identify all the node pairs of faces B_1^- and B_1^+ and to apply the MPC between the nodes from the same node pair. To easily manage the application of the macroscopic loading and, more interestingly, to automatize the determination of the macroscopic mechanical response, a reference point (using the ABAQUS terminology), designated RP_1, is created. The macroscopic loading is applied by imposing the following displacement on RP_1: $\left(U_1 = \left(F_{11} - 1 \right) \lambda_0; U_2 = 0; U_3 = 0 \right)$. This reference point is connected with each node pair to apply the MPC represented by Eq. (32). The reaction forces induced by the displacement applied on RP_1 are equal to the components 11, 12 and 13 of the macroscopic Piola–Kirchhoff stress tensor $\mathbf{P}$ multiplied by the initial volume of the unit cell $|\mathcal{V}_0|$ [21]. Quite similar developments can be performed for the application of the periodic boundary conditions on the other faces.

4 Computation of the macroscopic tangent modulus

The practical aspects related to the solution of the periodic homogenization problem by the finite element method have been detailed in Sect. 3. This solution is conducted by using the *Homtools* capable of handling the application of the periodic boundary conditions and the macroscopic loading as well as the determination of the macroscopic response. To achieve this technical effort, we have developed a set of python scripts to automatically determine the numerical evolution of the macroscopic tangent modulus by using three FEM-based techniques. This is the main objective of the present section, where the theoretical concepts behind the three techniques are briefly revisited and the practical implementations of these techniques within ABAQUS/Standard FE code are extensively discussed.

4.1 Perturbation technique

4.1.1 Numerical concept

The macroscopic tangent modulus, denoted $\mathbf{C}^{(PK1)}$, links the rate of the macroscopic deformation gradient $\dot{\mathbf{F}}$ to the rate of the macroscopic first Piola–Kirchhoff stress tensor

$\dot{\mathbf{P}}$. Hence, $\mathbf{C}^{(PK1)}$ can be obtained by the differentiation of the first Piola–Kirchhoff stress tensor with respect to the deformation gradient:

$$\dot{\mathbf{P}} = \mathbf{C}^{(PK1)} : \dot{\mathbf{F}} \Leftrightarrow \Delta\mathbf{P} \approx \mathbf{C}^{(PK1)} : \Delta\mathbf{F}, \tag{33}$$

where $\Delta\bullet$ is a very small perturbation applied to field $\bullet$.

Following the perturbation technique, each column of $\mathbf{C}^{(PK1)}$ is numerically constructed by perturbing the components of the macroscopic deformation gradient $\mathbf{F}$ and using the associated perturbed response as follows:

$$\forall i,j,k,l = 1,2,3 : \quad C_{ijkl}^{(PK1)} \approx \frac{\Delta P_{ij}^{(\alpha)}}{\Delta F_{kl}^{(\alpha)}} \approx \frac{P_{ij}(\mathbf{F}_{kl}^{(\alpha)}) - P_{ij}(\mathbf{F})}{\alpha}$$

$$\text{with} \quad \mathbf{F}_{kl}^{(\alpha)} = \mathbf{F} + \Delta\mathbf{F}_{kl}^{(\alpha)} = \mathbf{F} + \alpha\,\vec{\mathbf{e}}_k \otimes \vec{\mathbf{e}}_l, \tag{34}$$

where $\vec{\mathbf{e}}_k$ and $\vec{\mathbf{e}}_l$ are respectively the kth and lth unit vector, and α is the perturbation magnitude (which typically ranges between 10^{-6} and 10^{-8}). For illustration, the matrix form of Eq. (34) corresponding to the perturbation tensor $\Delta\mathbf{F}_{11}^{(\alpha)} = \alpha\,\vec{\mathbf{e}}_1 \otimes \vec{\mathbf{e}}_1$ can be written as:

$$\begin{bmatrix} C_{1111}^{(PK1)} & C_{1122}^{(PK1)} & C_{1133}^{(PK1)} & C_{1112}^{(PK1)} & C_{1123}^{(PK1)} & C_{1113}^{(PK1)} & C_{1121}^{(PK1)} & C_{1132}^{(PK1)} & C_{1131}^{(PK1)} \\ C_{2211}^{(PK1)} & C_{2222}^{(PK1)} & C_{2233}^{(PK1)} & C_{2212}^{(PK1)} & C_{2223}^{(PK1)} & C_{2213}^{(PK1)} & C_{2221}^{(PK1)} & C_{2232}^{(PK1)} & C_{2232}^{(PK1)} \\ C_{3311}^{(PK1)} & C_{3322}^{(PK1)} & C_{3333}^{(PK1)} & C_{3312}^{(PK1)} & C_{3323}^{(PK1)} & C_{3313}^{(PK1)} & C_{3321}^{(PK1)} & C_{3332}^{(PK1)} & C_{3331}^{(PK1)} \\ C_{1211}^{(PK1)} & C_{1222}^{(PK1)} & C_{1233}^{(PK1)} & C_{1212}^{(PK1)} & C_{1223}^{(PK1)} & C_{1213}^{(PK1)} & C_{1221}^{(PK1)} & C_{1232}^{(PK1)} & C_{1231}^{(PK1)} \\ C_{2311}^{(PK1)} & C_{2322}^{(PK1)} & C_{2333}^{(PK1)} & C_{2312}^{(PK1)} & C_{2323}^{(PK1)} & C_{2313}^{(PK1)} & C_{2321}^{(PK1)} & C_{2332}^{(PK1)} & C_{2331}^{(PK1)} \\ C_{1311}^{(PK1)} & C_{1322}^{(PK1)} & C_{1333}^{(PK1)} & C_{1312}^{(PK1)} & C_{1323}^{(PK1)} & C_{1313}^{(PK1)} & C_{1321}^{(PK1)} & C_{1332}^{(PK1)} & C_{1331}^{(PK1)} \\ C_{2111}^{(PK1)} & C_{2122}^{(PK1)} & C_{2133}^{(PK1)} & C_{2112}^{(PK1)} & C_{2123}^{(PK1)} & C_{2113}^{(PK1)} & C_{2121}^{(PK1)} & C_{2132}^{(PK1)} & C_{2131}^{(PK1)} \\ C_{3211}^{(PK1)} & C_{3222}^{(PK1)} & C_{3233}^{(PK1)} & C_{3212}^{(PK1)} & C_{3223}^{(PK1)} & C_{3213}^{(PK1)} & C_{3221}^{(PK1)} & C_{3232}^{(PK1)} & C_{3231}^{(PK1)} \\ C_{3111}^{(PK1)} & C_{3122}^{(PK1)} & C_{3133}^{(PK1)} & C_{3112}^{(PK1)} & C_{3123}^{(PK1)} & C_{3113}^{(PK1)} & C_{3121}^{(PK1)} & C_{3132}^{(PK1)} & C_{3131}^{(PK1)} \end{bmatrix} \begin{Bmatrix} \alpha \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} \Delta P_{11}^{(\alpha)} \\ \Delta P_{22}^{(\alpha)} \\ \Delta P_{33}^{(\alpha)} \\ \Delta P_{12}^{(\alpha)} \\ \Delta P_{23}^{(\alpha)} \\ \Delta P_{13}^{(\alpha)} \\ \Delta P_{21}^{(\alpha)} \\ \Delta P_{32}^{(\alpha)} \\ \Delta P_{31}^{(\alpha)} \end{Bmatrix}. \tag{35}$$

Thus, the first column of modulus $\mathbf{C}^{(PK1)}$ is computed as:

$$\forall i, j = 1, 2, 3 : \quad C_{ij11}^{PK1} \approx \frac{P_{ij}(\mathbf{F}_{11}^{(\alpha)}) - P_{ij}(\mathbf{F})}{\alpha}. \tag{36}$$

The eight other columns are obtained in the same way as for the first column.

4.1.2 Practical implementation

To determine the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$ by the perturbation technique, ten FE computation steps are performed: one general computation step to compute the unperturbed tensor $\mathbf{P}(\mathbf{F})$, followed by nine perturbation steps to compute the perturbed tensors $\mathbf{P}\left(\mathbf{F}_{kl}^{(\alpha)}\right)$ for $k, l = 1, 2, 3$. The perturbation steps are achieved using the ABAQUS/Standard restart technique. In the general computation step, it is needed to specify the requested restart files in the input file. An analysis can be restarted only if the restart request is made for it. In ABAQUS/Standard, these files include the restart (.res), analysis database (.mdl and.stt), part (.prt), output database (.odb), and substructure database (.sim) files. Accordingly, based on these restart data at each specific restart time, the nine perturbation steps will restart the analysis with the perturbed macroscopic deformation gradient. It is worth noting that the general computation step can be linear or nonlinear, modeled via either built-in material model or user-defined material subroutine (UMAT). As to the perturbation computation steps, they are performed using the linear static perturbation analysis of ABAQUS/Standard. The algorithmic steps of the perturbation technique are illustrated in Fig. 2.

**Box 1.** Main steps for the implementation of the perturbation technique.

> The main steps for the implementation of the perturbation technique are summarized as follows:
>
> - **Step 1**: run the general static step with addition of the option '*Restart, write, frequency=n' in the '.inp' file and then compute the unperturbed stress $\mathbf{P}(\mathbf{F})$ by the *Homtools*. Integer n, used as frequency parameter, specifies the increments at which restart information will be written. For example, frequency=2 is to write restart information at increments 2, 4, 6, etc.
>
> - **Step 2**: create nine restart '.inp' files with the option '*RESTART, READ, STEP=1, INC=n, END STEP' to specify the step and the increment from which restart analysis continues (see Appendix B). In these files, static perturbation analysis step is used and the corresponding option is:
>
>   '*STEP, PERTURBATION
>
>   *Static'
>
> - **Step 3**: run these nine restart jobs with command 'abaqus job=job-name oldjob=oldjob-name', or 'abaqus job=job-name oldjob=oldjob-name user=umat.f' when a UMAT is used in Step 1.
>
> - **Step 4**: output $\mathbf{P}\left(\mathbf{F}_{kl}^{(\alpha)}\right)$ corresponding to $\mathbf{F}_{kl}^{(\alpha)}$ (k, l = 1, 2, 3) and construct the different columns of $\mathbf{C}^{(PK1)}$ by substituting $\mathbf{P}(\mathbf{F})$, computed in Step 1, and $\mathbf{P}\left(\mathbf{F}_{kl}^{(\alpha)}\right)$ into the differential rule of Eq. (35).
>
> We have developed a set of Python codes to automatically perform Steps 2→4. Our codes are devoted to managing the execution of the general and perturbed steps as well as the numerical construction of $\mathbf{C}^{(PK1)}$.

Fig. 2 Basic algorithmic steps for the perturbation technique
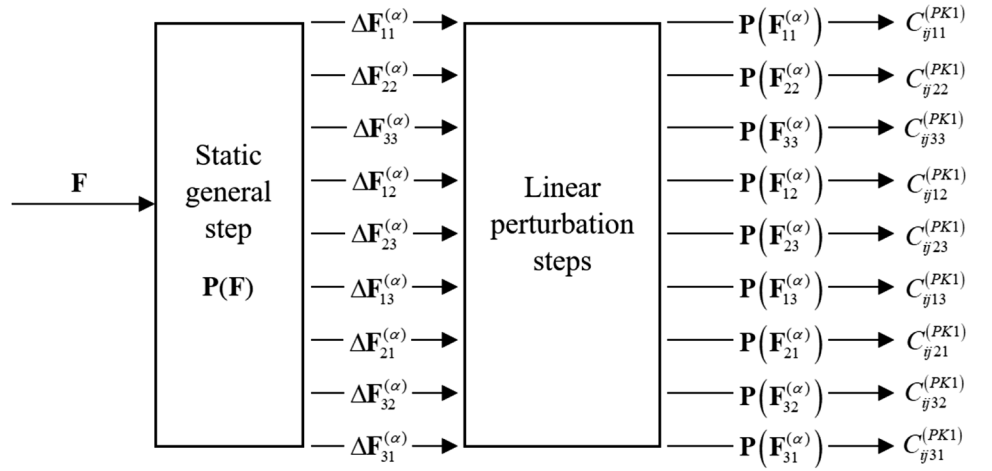
4.2 Condensation technique

4.2.1 Numerical concept

This technique is based on a condensation procedure of the global stiffness matrix $\mathbf{K}$ introduced in Eq. (10). As previously explained in Sect. 2, when the ABAQUS/Standard built-in material models are used to describe the microscopic behavior, the material part $\mathbf{K}_M$ of the stiffness matrix is constructed on the basis of the elastic tangent modulus. In this case, the condensation technique only gives the elastic macroscopic tangent modulus whether the mechanical behavior is elastic or inelastic. To avoid this problem, a UMAT should be used to implement the constitutive equations at the microscopic scale.

To apply the condensation technique, the nodes of the unit cell mesh shall be partitioned into two sets: set Λ and set Γ composed by the nodes in the interior and on the boundary of the unit cell, respectively. Let N_Λ and N_Γ denote the number of nodes of sets Λ and Γ, respectively. Following this partition, let us introduce the three link-topology matrices $\mathbb{D}_q \in \mathcal{R}^9 \times \mathcal{R}^3, \mathbb{Q}_q \in \mathcal{R}^9 \times \mathcal{R}^3$ and $\mathbb{H}_q \in \mathcal{R}^3 \times \mathcal{R}^{3N_\Gamma}$ defined in the subsequent developments. The use of these matrices enables to simplify the algorithmic treatment and implementation of the condensation technique.

The topology matrix $\mathbb{D}_q$, associated with node q that belongs to the boundary of the unit cell, consists of the components x_{q01}, x_{q02} and x_{q03} of the initial coordinate vector $\mathbf{x}_{q0}$ of node q as follows:

$$\forall q = 1, \dots, N_\Gamma : \quad \mathbb{D}_q = \begin{bmatrix} x_{q01} & 0 & 0 \\ 0 & x_{q02} & 0 \\ 0 & 0 & x_{q03} \\ x_{q02} & 0 & 0 \\ 0 & x_{q03} & 0 \\ x_{q03} & 0 & 0 \\ 0 & x_{q01} & 0 \\ 0 & 0 & x_{q02} \\ 0 & 0 & x_{q01} \end{bmatrix}. \tag{37}$$

Matrix $\mathbb{D}_q$ is introduced in order to rewrite Eq. (30) in a matrix form that is more suitable for algorithmic implementation:

$$\forall q = 1, \dots, N_p : \quad \mathbf{x}_{q^+} - \mathbf{x}_{q^-} = \left(\mathbb{D}_{q^+}^T - \mathbb{D}_{q^-}^T \right) \cdot {}^v\mathbf{F} := \mathbb{Q}_q^T \cdot {}^v\mathbf{F}, \tag{38}$$

with N_p being the total number of node pairs, and ${}^v\mathbf{F}$ the vector representation of the deformation gradient tensor $\mathbf{F}$.

As to matrix $\mathbb{H}_q$, it is built to link the current coordinate vector of the nodes on the boundary of the unit cell $\mathbf{x}_b$ to the current coordinate vector $\mathbf{x}_q = \mathbf{x}_{q^+} - \mathbf{x}_{q^-}$ of node pair q:

$$\forall q = 1, \dots, N_p : \quad \mathbf{x}_q = \mathbb{H}_q \cdot \mathbf{x}_b. \tag{39}$$

The components of matrix $\mathbb{H}_q$ take one of the following values: $0, -1$ or 1.

Using matrices $\mathbb{H}_q$ and $\mathbb{Q}_q$, Eq. (38) can be re-expressed in a more compact form:

$$\forall q = 1, \dots, N_p : \quad \mathbb{H}_q \cdot \mathbf{x}_b = \mathbb{Q}_q^T \cdot {}^v\mathbf{F}. \tag{40}$$

Thus, all the periodic constraints of the unit cell can be assembled in the global notation:

$$\mathbb{H} \cdot \mathbf{x}_b = \mathbb{Q}^T \cdot {}^v\mathbf{F}, \tag{41}$$

where global matrices $\mathbb{H}$ and $\mathbb{Q}$ are constructed by concatenation of their nodal counterparts $\mathbb{H}_q$ and $\mathbb{Q}_q$ $(1 \leq q \leq N_p)$:

$$\mathbb{H} := \begin{bmatrix} \mathbb{H}_1 \\ \vdots \\ \mathbb{H}_q \\ \vdots \\ \mathbb{H}_{N_p} \end{bmatrix} \quad ; \quad \mathbb{Q}^T := \begin{bmatrix} \mathbb{Q}_1^T \\ \vdots \\ \mathbb{Q}_q^T \\ \vdots \\ \mathbb{Q}_{N_p}^T \end{bmatrix}. \tag{42}$$

On the other hand, following the partition of the whole node set into subsets Λ and Γ, Eq. (10) can be restated in the following form (after permutation of lines and columns
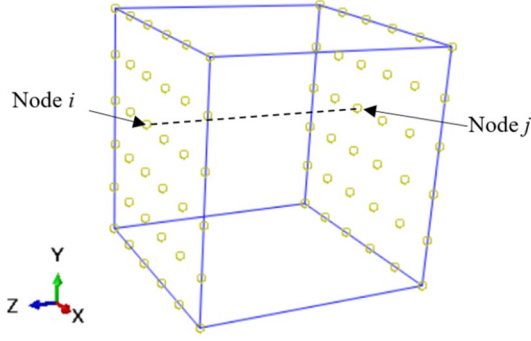
Fig. 3 Node pairs in Z-direction

of matrix $\mathbf{K}$ and the corresponding components of vectors $\dot{\mathbf{q}}$ and $\dot{\mathbf{r}}$):

$$\begin{bmatrix} \mathbf{K}_{\Lambda\Lambda} & \mathbf{K}_{\Lambda\Gamma} \\ \mathbf{K}_{\Gamma\Lambda} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \cdot \left\{ \begin{array}{c} \dot{\mathbf{q}}_\Lambda \\ \dot{\mathbf{q}}_\Gamma \end{array} \right\} = \left\{ \begin{array}{c} \dot{\mathbf{r}}_\Lambda \\ \dot{\mathbf{r}}_\Gamma \end{array} \right\}. \tag{43}$$

As a consequence of the microscopic equilibrium equation (23), the internal force vector $\dot{\mathbf{r}}_\Lambda$ is equal to $\mathbf{0}$ at the convergence of the FE computations. Hence, Eq. (43) becomes:

$$\begin{bmatrix} \mathbf{K}_{\Lambda\Lambda} & \mathbf{K}_{\Lambda\Gamma} \\ \mathbf{K}_{\Gamma\Lambda} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \cdot \left\{ \begin{array}{c} \dot{\mathbf{q}}_\Lambda \\ \dot{\mathbf{q}}_\Gamma \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{0} \\ \dot{\mathbf{r}}_\Gamma \end{array} \right\}. \tag{44}$$

The elimination of $\dot{\mathbf{q}}_\Lambda$ from Eq. (44) allows deriving the condensed (or reduced) stiffness matrix $\tilde{\mathbf{K}}_{\Gamma\Gamma}$ relating the boundary displacement rate $\dot{\mathbf{q}}_\Gamma$ to $\dot{\mathbf{r}}_\Gamma$:

$$\tilde{\mathbf{K}}_{\Gamma\Gamma} \cdot \dot{\mathbf{q}}_\Gamma = \dot{\mathbf{r}}_\Gamma \quad \text{with} \quad \tilde{\mathbf{K}}_{\Gamma\Gamma} = \mathbf{K}_{\Gamma\Gamma} - \mathbf{K}_{\Gamma\Lambda} \cdot \mathbf{K}_{\Lambda\Lambda}^{-1} \cdot \mathbf{K}_{\Lambda\Gamma}. \tag{45}$$

At the convergence of incremental analysis, the matrix form of the macroscopic tangent modulus $^m\mathbf{C}^{(PK1)}$ is computed in terms of the condensed stiffness matrix $\tilde{\mathbf{K}}_{\Gamma\Gamma}$, matrices $\mathbb{H}$ and $\mathbb{Q}$ as follows [17, 43]:

$$^m\mathbf{C}^{(PK1)} = \frac{1}{|\mathcal{V}_0|} \mathbb{Q} \cdot \left[\mathbb{H} \cdot \tilde{\mathbf{K}}_{\Gamma\Gamma}^{-1} \cdot \mathbb{H}^T \right]^{-1} \cdot \mathbb{Q}^T. \tag{46}$$

Considering expression (45) of $\tilde{\mathbf{K}}_{\Gamma\Gamma}$, the matrix form of the macroscopic tangent modulus $^m\mathbf{C}^{(PK1)}$ can be expressed as:

$$^m\mathbf{C}^{(PK1)} = \frac{1}{|\mathcal{V}_0|} \mathbb{Q} \cdot \left[\mathbb{H} \cdot \left(\mathbf{K}_{\Gamma\Gamma} - \mathbf{K}_{\Gamma\Lambda} \cdot \mathbf{K}_{\Lambda\Lambda}^{-1} \cdot \mathbf{K}_{\Lambda\Gamma} \right)^{-1} \cdot \mathbb{H}^T \right]^{-1} \cdot \mathbb{Q}^T. \tag{47}$$

For the sake of brevity, only the final result of the condensation technique is given in Eq. (47). The interested readers may refer to references [17] and [43] for further details on how Eq. (47) is obtained. Analogous developments will be carried out in Sect. 5 to determine the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$ by the condensation method for the

particular case of symmetric problems (which is the main theoretical novelty of the present contribution).

4.2.2 Practical implementation

As presented in Sect. 4.2.1, the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$ is obtained by a condensation procedure of the global stiffness matrix $\mathbf{K}$, which is assembled from the elementary stiffness matrices $\mathbf{K}_{el}$. The requested option to output the elementary stiffness matrices is '*Element Matrix Output', which needs to be added in the input file. To illustrate the operational aspects, a brief example of an input file is provided in Appendix A. After achieving the finite element computation, the elementary stiffness matrices will be stored in an external file with '.mtx' extension (e.g. 'myMatrix.mtx' as in the example of Appendix A). This file is the input of the Python codes developed to assemble and handle the global stiffness matrix $\mathbf{K}$. Recalling that all the nodes of the unit cell mesh are grouped into two sets Λ and Γ; hence, a node q belongs to set Γ if at least one of the following conditions holds:

$$x_{q\,01} = -\lambda_0/2 \text{ or } \lambda_0/2; \quad x_{q\,02} = -\lambda_0/2 \text{ or } \lambda_0/2;$$
$$x_{q\,03} = -\lambda_0/2 \text{ or } \lambda_0/2, \tag{48}$$

otherwise, it belongs to set Λ.

In order to construct topology matrices $\mathbb{Q}$ and $\mathbb{H}$, the set of node pairs as well as the corresponding initial coordinates need to be identified. As illustrated in Fig. 3, nodes i and j make up a node pair in the Z-direction.

To identify all the node pairs in node set Γ, the following algorithm is developed and implemented into the Python code.

Nodes i and j represent a node pair in X-direction, if:
$x_{i\,01} - x_{j\,01} = \pm\lambda_0$ and $x_{i\,02} = x_{j\,02}$ and $x_{i\,03} = x_{j\,03}$.
Nodes i and j represent a node pair in Y-direction, if:
$x_{i\,02} - x_{j\,02} = \pm\lambda_0$ and $x_{i\,01} = x_{j\,01}$ and $x_{i\,03} = x_{j\,03}$.
Nodes i and j represent a node pair in Z-direction, if:
$x_{i\,03} - x_{j\,03} = \pm\lambda_0$ and $x_{i\,01} = x_{j\,01}$ and $x_{i\,02} = x_{j\,02}$.
The components of matrix $\mathbb{H} \in \mathcal{R}^{3N_P} \times \mathcal{R}^{3N_\Gamma}$ take one of three possible values: 0, -1 or 1. These components can be determined as follows, once all the node pairs detected:

If Nodes i and j form a node pair (with $i < j$), the distribution of values 1 and -1 in $\mathbb{H}$ can be summarized as:

$$\mathbb{H}_{3i-2,3i-2} = 1; \quad \mathbb{H}_{3i-2,3j-2} = -1,$$
$$\mathbb{H}_{3i-1,3i-1} = 1; \quad \mathbb{H}_{3i-1,3j-1} = -1, \tag{49}$$
$$\mathbb{H}_{3i,3i} = 1; \quad \mathbb{H}_{3i,3j} = -1.$$

The other components of matrix $\mathbb{H}$, not defined by Eq. (49), are set to 0.

Matrix $\mathbb{D} \in \mathcal{R}^9 \times \mathcal{R}^{3N_\Gamma}$ is assembled from matrices $\mathbb{D}_q \in \mathcal{R}^9 \times \mathcal{R}^3$, defined in Eq. (37), as follows:

$$\mathbb{D} = \begin{bmatrix} \cdots & \overbrace{\begin{matrix} x_{i01} & 0 & 0 \\ 0 & x_{i02} & 0 \\ 0 & 0 & x_{i03} \\ x_{i02} & 0 & 0 \\ 0 & x_{i03} & 0 \\ x_{i03} & 0 & 0 \\ 0 & x_{i01} & 0 \\ 0 & 0 & x_{i02} \\ 0 & 0 & x_{i01} \end{matrix}}^{\mathbb{D}_i} & \cdots & \overbrace{\begin{matrix} x_{j01} & 0 & 0 \\ 0 & x_{j02} & 0 \\ 0 & 0 & x_{j03} \\ x_{j02} & 0 & 0 \\ 0 & x_{j03} & 0 \\ x_{j03} & 0 & 0 \\ 0 & x_{j01} & 0 \\ 0 & 0 & x_{j02} \\ 0 & 0 & x_{j01} \end{matrix}}^{\mathbb{D}_j} & \cdots \end{bmatrix}. \tag{50}$$

Once matrix $\mathbb{D}$ constructed, matrix $\mathbb{Q}$ can be determined by the following matrix multiplication:

$$\mathbb{Q} = \mathbb{D} \cdot \mathbb{H}^T. \tag{51}$$

With the labels of nodes belonging to sets Λ and Γ, one can easily extract the four submatrices $\mathbf{K}_{\Lambda\Lambda}$, $\mathbf{K}_{\Lambda\Gamma}$, $\mathbf{K}_{\Gamma\Lambda}$ and $\mathbf{K}_{\Gamma\Gamma}$ from the global matrix $\mathbf{K}$, as stated in Eq. (43). Thus far, all the ingredients needed to compute $\mathbf{C}^{(PK1)}$ are prepared.

4.3 Fluctuation technique

4.3.1 Numerical concept

Considering Eqs. (26) and (27), the following condition should be fulfilled at the convergence of the finite element iterations:

$$G := \int_{\mathcal{V}_0} \left[\frac{\partial \delta \mathbf{v}_{per}}{\partial \mathbf{x}_0} : \dot{\mathbf{p}} \right] d\mathcal{V}_0 = 0. \tag{52}$$

The linearization of Eq. (52), classically used in finite element computations, takes the form:

$$G + \mathcal{D}G = 0, \tag{53}$$

where $\mathcal{D}\bullet$ is the iteration of field $\bullet$ (in the finite element sense). Higher-order terms are neglected in the linearized form given by Eq. (53).

The linearized form (53) requires that $G \to 0$ and $\mathcal{D}G \to 0$ at the convergence of the finite element computation. Hence, the following iterative form can be derived from Eq. (52):

Box 2. Main steps for the implementation of the condensation technique.

The main steps for the implementation of the condensation technique are summarized as follows:

- **Step 1**: add the option '*Element Matrix Output' in the input file; run the finite element computation with a user subroutine UMAT, the elementary stiffness matrices $\mathbf{K}_{el}$ will be outputted in a '.mtx' file.

- **Step 2**: assemble the global stiffness matrix $\mathbf{K}$ from elementary stiffness matrices stored in the '.mtx' file by using the connectivity of the different nodes of the elements; partition $\mathbf{K}$ into four submatrices $\mathbf{K}_{\Lambda\Lambda}$, $\mathbf{K}_{\Lambda\Gamma}$, $\mathbf{K}_{\Gamma\Lambda}$ and $\mathbf{K}_{\Gamma\Gamma}$, after permutation of its lines and columns as stated in Eq. (44).

- **Step 3**: compute the condensed matrix $\tilde{\mathbf{K}}_{\Gamma\Gamma}$ from submatrices $\mathbf{K}_{\Lambda\Lambda}$, $\mathbf{K}_{\Lambda\Gamma}$, $\mathbf{K}_{\Gamma\Lambda}$ and $\mathbf{K}_{\Gamma\Gamma}$ on the basis of Eq. (46).

- **Step 4**: construct matrices $\mathbb{H}$ and $\mathbb{Q}$ by using Eqs. (50) and (52), respectively.

- **Step 5**: compute the matrix form of the macroscopic tangent modulus $^m\mathbf{C}^{(PK1)}$ by using Eq. (48). Finally, deduce the fourth-order tensor $\mathbf{C}^{(PK1)}$.

We have developed a set of Python scripts to automatically manage Steps 2→5. To improve the efficiency of the condensation method, the developed scripts are parallelized with multiple processors (see Appendix B for more details). The execution of Step 5 inflicts high CPU and memory costs. These costs are mainly due to the double matrix-inversion required in Eq. (48). To efficiently optimize this execution, these inversion operations are performed by using the functions 'linalg.solve' and 'linalg.pinvh' of the NumPy library.

$$\mathcal{D}G = \int_{\mathcal{V}_0} \left[\frac{\partial\, \delta \mathbf{v}_{per}}{\partial\, \mathbf{x}_0} : \mathbf{c}^{(PK1)} : \left(\mathcal{D}\dot{\mathbf{F}} + \mathcal{D}\frac{\partial\, \mathbf{v}_{per}}{\partial\, \mathbf{x}_0} \right) \right] \mathrm{d}\mathcal{V}_0 = 0.$$
(54)

After finite element discretization, the periodic velocity $\mathbf{v}_{per}$ can be obtained from the interpolation rule (similar to the one used in Eq. (8)):

$$\mathbf{v}_{per} = \mathbf{N} \cdot \dot{\mathbf{q}}_{per}.$$
(55)

The substitution of Eq. (55) into Eq. (54) leads to the following linear algebraic system:

$$\mathbf{K} \cdot \mathcal{D}\dot{\mathbf{q}}_{\mathrm{per}} = -\hat{\mathbf{K}} \cdot \mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right),$$
(56)

where the global stiffness matrix $\mathbf{K}$ and a fluctuation matrix $\hat{\mathbf{K}}$ are defined as:

$$\mathbf{K} = \bigcup_{el=1}^{n} \int_{\mathcal{V}_0^{el}} \mathbf{B}^T \cdot {}^{m}\mathbf{c}^{(PK1)} \cdot \mathbf{B} \mathrm{d}\mathcal{V}_0^{el} \quad ; \quad \hat{\mathbf{K}} = \bigcup_{el=1}^{n} \int_{\mathcal{V}_0^{el}} \mathbf{B}^T \cdot {}^{m}\mathbf{c}^{(PK1)} \mathrm{d}\mathcal{V}_0^{el},$$
(57)

where $\bigcup_{el=1}^{n}$ denotes the finite element assembly operator when the unit cell is discretized by n finite elements el. It is worth noting that the global stiffness matrix $\mathbf{K}$ defined by Eq. (57)$_1$ is exactly the same as the one introduced in Eq. (10) and equal to $\left[\mathbf{K}_M + \mathbf{K}_{GNL}\right]$ (as long as the relations between the microscopic tangent moduli $\mathbf{c}^{(0)}$, $\mathbf{c}^{(2)}$, $\mathbf{c}^{(PK1)}$ and $\mathbf{c}^{(PK2)}$ are correctly defined).

Vector $\mathcal{D}\dot{\mathbf{q}}_{\mathrm{per}}$ can be obtained by solving Eq. (56):

$$\mathcal{D}\dot{\mathbf{q}}_{per} = -\mathbf{K}^{-1} \cdot \hat{\mathbf{K}} \cdot \mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right),$$
(58)

and then iteration $\mathcal{D}\left({}^{v}\dot{\mathbf{f}}_{per}\right)$ can be obtained from Eq. (58) by:

$$\mathcal{D}\left({}^{v}\dot{\mathbf{f}}_{per}\right) = \mathbf{B} \cdot \mathcal{D}\dot{\mathbf{q}}_{per} = -\mathbf{B} \cdot \mathbf{K}^{-1} \cdot \hat{\mathbf{K}} \cdot \mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right).$$
(59)

On the other hand, we have the following relation:

$$\mathcal{D}\left({}^{v}\dot{\mathbf{P}}\right) = \frac{1}{|\mathcal{V}_0|} \int_{\mathcal{V}_0} \mathcal{D}({}^{v}\dot{\mathbf{p}})\, \mathrm{d}\mathcal{V}_0$$
$$= \frac{1}{|\mathcal{V}_0|} \int_{\mathcal{V}_0} {}^{m}\mathbf{c}^{(PK1)} \cdot \left(\mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right) + \mathcal{D}\left({}^{v}\dot{\mathbf{f}}_{per}\right) \right) \mathrm{d}\mathcal{V}_0.$$
(60)

The combination of Eqs. (59) and (60) yields:

$$\mathcal{D}\left({}^{v}\dot{\mathbf{P}}\right) = \frac{1}{|\mathcal{V}_0|} \int_{\mathcal{V}_0} \mathcal{D}({}^{v}\dot{\mathbf{p}})\, \mathrm{d}\mathcal{V}_0$$
$$= \frac{1}{|\mathcal{V}_0|} \int_{\mathcal{V}_0} {}^{m}\mathbf{c}^{(PK1)} \cdot \left(\left({}^{v}\dot{\mathbf{F}}\right) - \mathbf{B} \cdot \mathbf{K}^{-1} \cdot \hat{\mathbf{K}} \cdot \left({}^{v}\dot{\mathbf{F}}\right) \right) \mathrm{d}\mathcal{V}_0.$$
(61)

As iteration $\mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right)$ is homogeneous over the volume $\mathcal{V}_0$ of the unit cell, Eq. (61) can be reformulated as:

$$\mathcal{D}\left({}^{v}\dot{\mathbf{P}}\right) = \frac{1}{|\mathcal{V}_0|} \int_{\mathcal{V}_0} \mathcal{D}({}^{v}\dot{\mathbf{p}})\, \mathrm{d}\mathcal{V}_0$$
$$= \frac{1}{|\mathcal{V}_0|} \left(\int_{\mathcal{V}_0} {}^{m}\mathbf{c}^{(PK1)} \mathrm{d}\mathcal{V}_0 - \hat{\mathbf{K}}^T \cdot \mathbf{K}^{-1} \cdot \hat{\mathbf{K}} \right) \cdot \mathcal{D}\left({}^{v}\dot{\mathbf{F}}\right).$$
(62)

By comparing Eqs. (33) and (62), one can easily deduce that:

$$^{m}\mathbf{C}^{(PK1)} = \frac{1}{|\mathcal{V}_0|} \left(\int_{\mathcal{V}_0} {}^{m}\mathbf{c}^{(PK1)} \mathrm{d}\mathcal{V}_0 - \hat{\mathbf{K}}^T \cdot \mathbf{K}^{-1} \cdot \hat{\mathbf{K}} \right).$$
(63)

4.3.2 Practical implementation

As stated by Eq. (63), the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$ consists of two main parts: the volume average of the microscopic moduli $\mathbf{c}^{(PK1)}$ and a fluctuation part dependent on the global stiffness matrix $\mathbf{K}$ and on a global fluctuation matrix $\hat{\mathbf{K}}$. Global matrices $\mathbf{K}$ and $\hat{\mathbf{K}}$ are obtained from their elementary counterparts $\mathbf{K}_{el}$ and $\hat{\mathbf{K}}_{el}$ by the assembly rules given by Eq. (57). Elementary matrices $\mathbf{K}_{el}$ and $\hat{\mathbf{K}}_{el}$ are dependent on $\mathbf{c}^{(PK1)}$. Hence, to ensure the accurate computation of the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$, the microscopic tangent moduli $\mathbf{c}^{(PK1)}$ should be correctly defined and implemented. Furthermore, to determine the different ingredients required for the computation of $\mathbf{C}^{(PK1)}$ by the fluctuation technique, a user element (UEL) subroutine needs to be used. To achieve this task, we have used the UEL developed in [44], after some modifications and additions. Indeed, the virtual work principle has been slightly modified to consider the Jaumann derivative of the Kirchhoff stress instead of the Truesdell derivative used in the initial version. After these modifications, the elementary stiffness matrix $\mathbf{K}_{el}$ has been correctly computed in the UEL by adding the contribution of the geometric nonlinearities $\mathbf{K}_{GNL\,el}$ (see Eq. (9)$_2$) to the contribution due to material behavior $\mathbf{K}_{M\,el}$ (see Eq. (9)$_1$). As to $\mathbf{c}^{(PK1)}$, which is associated with each integration point, it is determined from the Jacobian matrix **DDSDDE** (an output of the corresponding UMAT) by using Eqs. (7), (16) and (17). Once $\mathbf{c}^{(PK1)}$ are determined for all the integration points of the element, the elementary contribution to the volume average of the microscopic moduli $\int_{\mathcal{V}_0^{el}} \mathbf{c}^{(PK1)} \mathrm{d}\mathcal{V}_0^{el}$ can be determined by an easy integration operation, and the elementary stiffness matrix $\mathbf{K}_{el}$ and fluctuation matrix $\hat{\mathbf{K}}_{el}$ can be computed by the following relations:

$$\mathbf{K}_{el} = \int_{\mathcal{V}_0^{el}} \mathbf{B}^T \cdot {}^{m}\mathbf{c}^{(PK1)} \cdot \mathbf{B}\, \mathrm{d}\mathcal{V}_0^{el} \quad ; \quad \hat{\mathbf{K}}_{el} = \int_{\mathcal{V}_0^{el}} \mathbf{B}^T \cdot {}^{m}\mathbf{c}^{(PK1)} \mathrm{d}\mathcal{V}_0^{el}.$$
(64)

The different elementary contributions should be outputted and stored in separate files.

Fig. 4 RUC occupying the initial domain $\mathcal{V}_{r0} = [0, \lambda_0/2] \times [0, \lambda_0/2] \times [0, \lambda_0/2]$ and its corresponding FUC assumed to be symmetric about three planes of symmetry ($x_{01} = 0$, $x_{02} = 0$ and $x_{03} = 0$)
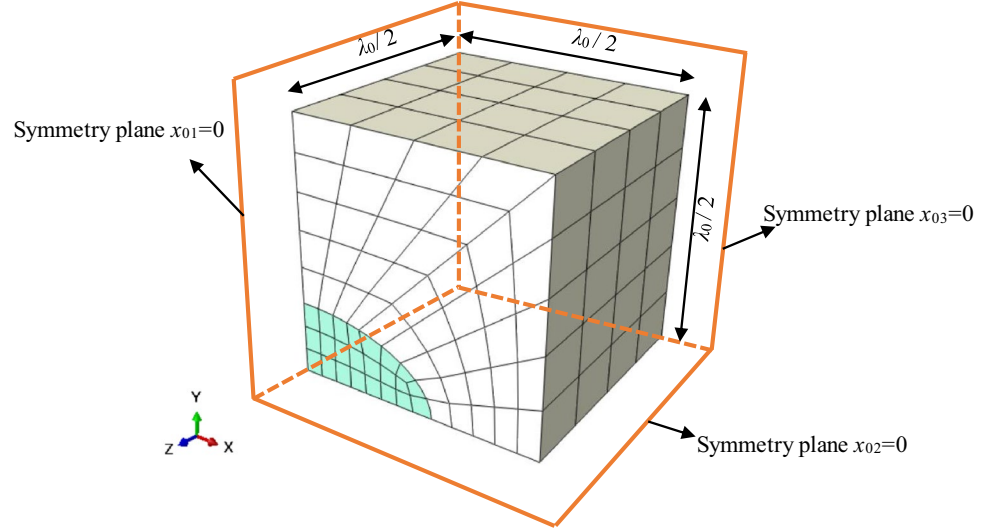
Box 3. Main steps for the implementation of the fluctuation technique.

The main steps for the implementation of the fluctuation technique are summarized as follows:

- **Step 1**: compute the microscopic tangent moduli $\mathbf{c}^{(PK1)}$ from jacobian matrix **DDSDDE** by using Eqs. (7), (16) and (17).

- **Step 2**: compute the elementary contributions (namely, $\mathbf{K}_{el}$, $\hat{\mathbf{K}}_{el}$ and $\int_{\mathcal{V}_0^{el}} \mathbf{c}^{(PK1)} \, d\mathcal{V}_0^{el}$) and store them in external files.

- **Step 3**: read the external files and use the connectivity of the different nodes to assemble the various elementary contributions and to obtain the global counterparts.

- **Step 4**: once the global counterparts determined, the matrix form of the macroscopic tangent modulus $^m\mathbf{C}^{(PK1)}$ can be computed by Eq. (63). Once $^m\mathbf{C}^{(PK1)}$ determined, the fourth-order tensor $\mathbf{C}^{(PK1)}$ can be easily deduced.

A set of Python codes has been developed to perform Steps 3→4. The algorithm of the codes can be found in Appendix B.

5 Extension to symmetric microstructures

In some applications, the studied heterogeneous media are made of unit cells exhibiting symmetry properties (composite materials, porous media…). Hence, it is essential to examine whether these symmetry properties can be efficiently exploited to speed up the computation of the overall properties of heterogeneous media. A heterogeneous unit cell has symmetry properties if the geometric distribution of the different phases is symmetric (see Fig. 4) and the behavior of each phase exhibits material symmetries (isotropic or orthotropic). Without loss of generality, we consider a 3D unit cell occupying the initial domain $\mathcal{V}_0 = [-\lambda_0/2, \lambda_0/2] \times [-\lambda_0/2, \lambda_0/2] \times [-\lambda_0/2, \lambda_0/2]$. This unit cell is assumed to be symmetric about three planes of symmetry ($x_{01} = 0$, $x_{02} = 0$ and $x_{03} = 0$), as displayed in Fig. 4. In this case, the study of one eighth of the unit cell, occupying the initial domain

$\mathcal{V}_{r0} = [0,\,\lambda_0/2] \times [0,\,\lambda_0/2] \times [0,\,\lambda_0/2]$, is sufficient to determine the mechanical response of the full unit cell only if the boundary conditions on the planes of symmetry ($x_{01} = 0$, $x_{02} = 0$ and $x_{03} = 0$) and on the outer surfaces ($x_{01} = \lambda_0/2$, $x_{02} = \lambda_0/2$ and $x_{03} = \lambda_0/2$) are correctly defined. Otherwise, the results of such analyses could mislead. The use of one eighth of the unit cell (shortly called RUC, as reduced unit cell), instead of the full one (denoted as FUC), allows dividing the number of mesh elements by 8 (for the same computation accuracy), thus considerably reducing the CPU time required for the computation of the macroscopic tangent modulus. The current section is focused on the theoretical extension of the perturbation and condensation techniques for the determination of the overall moduli of RUCs. This extension is valid only if the studied RUCs are subjected to simple loading histories (linear and basic strain paths, such as diagonal strain paths or simple shear loadings). For more general loading histories, the use of the RUC model leads to some inaccuracies in the prediction of the shear components of the tangent modulus $\mathbf{C}^{(PK1)}$. The practical aspects related to the implementation of these extended techniques in the particular case of RUCs are quite similar to those developed for the general case (Sects. 4.1.2, 4.2.2). These practical aspects are omitted for the sake of brevity.

5.1 Perturbation technique

The suitable boundary conditions to be applied on the RUC to ensure the same mechanical response as that of the whole unit cell have been initially established by Léné [45] for media undergoing small strain. The developments achieved by Léné [45] are extended in the present contribution to the finite strain framework, and these extensions are used to construct the overall tangent modulus by the perturbation technique. The prescribed boundary conditions, under the perturbed steps, are dependent on the applied perturbation. The details are given in Table 1 (with α being the magnitude of the perturbation introduced in Sect. 4.1.1). As shown in this table, the boundary conditions applied on the different DOFs are categorized into two main families:

- DOFs subjected to a small linear displacement increment $\Delta u_i = \left(\Delta \mathbf{F}_{kl}^{(\alpha)} \cdot \mathbf{x}_0 \right)_i$ (where $\mathbf{x}_0$ is the initial position of the associated node and $\left(\Delta \mathbf{F}_{kl}^{(\alpha)} \cdot \mathbf{x}_0 \right)_i$ is the ith component of vector $\Delta \mathbf{F}_{kl}^{(\alpha)} \cdot \mathbf{x}_0$);
- DOFs free from any displacement constraint (Δu_i free). In this case, the associated reaction force is obviously equal to zero.

To better explain these boundary conditions, let us detail them for the perturbation $\Delta \mathbf{F}_{11}^{(\alpha)}$:

- On face $x_{01} = 0$: $\Delta u_1 = \left(\Delta \mathbf{F}_{11}^{(\alpha)} \cdot \mathbf{x}_0 \right)_1 = 0$, while the other DOFs are free.
- On face $x_{01} = \lambda_0/2$: $\Delta u_1 = \left(\Delta \mathbf{F}_{11}^{(\alpha)} \cdot \mathbf{x}_0 \right)_1 = \alpha\lambda_0/2$, while the other DOFs are free.
- On faces $x_{02} = 0$ and $\lambda_0/2$: $\Delta u_2 = \left(\Delta \mathbf{F}_{11}^{(\alpha)} \cdot \mathbf{x}_0 \right)_2 = 0$, while the other DOFs are free.
- On faces $x_{03} = 0$ and $\lambda_0/2$: $\Delta u_3 = \left(\Delta \mathbf{F}_{11}^{(\alpha)} \cdot \mathbf{x}_0 \right)_3 = 0$, while the other DOFs are free.

The perturbed stress tensors $\mathbf{P}\left(\mathbf{F}_{kl}^{(\alpha)} \right)$, corresponding to the various perturbations $\Delta \mathbf{F}_{kl}^{(\alpha)}$ and required to construct the macroscopic tangent modulus $\mathbf{C}^{(PK1)}$ column by column, are derived from the reaction forces applied on the boundary of the reduced unit cell $\mathcal{S}_{r0}$:

$$\mathbf{P}\left(\mathbf{F}_{kl}^{(\alpha)} \right) = \frac{1}{|\mathcal{V}_{r0}|} \int_{\mathcal{V}_{r0}} \mathbf{p}\left(\mathbf{F}_{kl}^{(\alpha)} \right) \mathrm{d}\mathcal{V}_{r0} = \frac{1}{|\mathcal{V}_{r0}|} \int_{\mathcal{S}_{r0}} \mathbf{t}_0\left(\mathbf{F}_{kl}^{(\alpha)} \right) \otimes \mathbf{x}_0 \, \mathrm{d}\mathcal{S}_{r0}.$$
(65)

To define the different perturbation steps, the nine restart '.inp' files described in Step 2 of Box 1 need to be modified to integrate the difference in the boundary conditions from one perturbation step to another.

5.2 Condensation technique

The development of a condensation technique pertaining to unit cells exhibiting symmetry properties is presented hereafter. This development follows the general idea behind the same technique for full unit cells (without symmetry properties), summarized in Sect. 4.2.1, with some main adaptations that are essential to account for the particularities of RUCs. The perturbation technique presented in Sect. 5.1 is of great use to ensure the theoretical extension of the condensation technique. Unlike the original condensation technique, where the different columns of the overall tangent modulus $^m\mathbf{C}^{(PK1)}$ are simultaneously determined in a single matrix operation, the columns of the tangent modulus corresponding to the RUC are computed separately, through successive iterations. Indeed, the starting point of the original condensation technique consists in the partition of nodes into two different sets Λ and Γ. The composition of these node sets is independent of the boundary conditions applied on the unit cell. Then, the same sets can be used to compute all the columns of $^m\mathbf{C}^{(PK1)}$. For the extended condensation technique, the partition of DOF sets depends on the boundary conditions, as shown in Table 1. Therefore, it is not possible to simultaneously construct all the columns of $^m\mathbf{C}^{(PK1)}$ by using the same DOF partition, and thus the computation of

Table 1 Boundary conditions applied on the RUC

Perturbed components	Boundary conditions		
	faces $\quad x_{01}=0$ and $\lambda_0/2$	faces $\quad x_{02}=0$ and $\lambda_0/2$	faces $\quad x_{03}=0$ and $\lambda_0/2$
$kk=11\,;\,22\,;\,33$	$\Delta u_1 = \left(\Delta \mathbf{F}_{kk}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; Δu_3 free	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{kk}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free	Δu_1 free ; Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{kk}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$
12	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{12}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ $\Delta u_3 = \left(\Delta \mathbf{F}_{12}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	$\Delta u_1 = \left(\Delta \mathbf{F}_{12}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{12}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	Δu_1 free ; Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{12}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$
21	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{21}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ $\Delta u_3 = \left(\Delta \mathbf{F}_{21}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	$\Delta u_1 = \left(\Delta \mathbf{F}_{21}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{21}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	Δu_1 free ; Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{21}^{(\alpha)} \cdot \mathbf{x}_0\right)_3\,;$
23	$\Delta u_1 = \left(\Delta \mathbf{F}_{23}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; Δu_3 free	$\Delta u_1 = \left(\Delta \mathbf{F}_{23}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\;;$ Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{23}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	$\Delta u_1 = \left(\Delta \mathbf{F}_{23}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ $\Delta u_2 = \left(\Delta \mathbf{F}_{23}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free
32	$\Delta u_1 = \left(\Delta \mathbf{F}_{32}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; Δu_3 free	$\Delta u_1 = \left(\Delta \mathbf{F}_{32}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ Δu_2 free ; $\Delta u_3 = \left(\Delta \mathbf{F}_{32}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	$\Delta u_1 = \left(\Delta \mathbf{F}_{32}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ $\Delta u_2 = \left(\Delta \mathbf{F}_{32}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free
13	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{13}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ $\Delta u_3 = \left(\Delta \mathbf{F}_{13}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{13}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free	$\Delta u_1 = \left(\Delta \mathbf{F}_{13}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ $\Delta u_2 = \left(\Delta \mathbf{F}_{13}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free
31	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{31}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ $\Delta u_3 = \left(\Delta \mathbf{F}_{31}^{(\alpha)} \cdot \mathbf{x}_0\right)_3$	Δu_1 free ; $\Delta u_2 = \left(\Delta \mathbf{F}_{31}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free	$\Delta u_1 = \left(\Delta \mathbf{F}_{31}^{(\alpha)} \cdot \mathbf{x}_0\right)_1\,;$ $\Delta u_2 = \left(\Delta \mathbf{F}_{31}^{(\alpha)} \cdot \mathbf{x}_0\right)_2\,;$ Δu_3 free

the columns of $^m\mathbf{C}^{(PK1)}$ in successive iterations (one iteration for each column) is unavoidable.

In what follows, we detail how a column of $^m\mathbf{C}^{(PK1)}$ can be determined, and the same algorithm could be used to compute the other columns.

As a starting point of this algorithm, the DOFs of the nodes of the RUC mesh should be partitioned into two sets:

- Set Ψ: the set of DOFs on which the reaction forces are zero. This set includes the DOFs of all the nodes in the interior of the RUC as well as the DOFs of the nodes on the boundary, which are free from any displacement

constraint (see Table 2 for illustration). Then, set Ψ is defined by the following equation at convergence:

$$\dot{\mathbf{r}}_\Psi = \mathbf{0}, \tag{66}$$

where $\dot{\mathbf{r}}_\Psi$ is the residual rate vector corresponding to DOFs in set Ψ.

- Set Ω: the set of DOFs on the boundary of the RUC that are subjected to a linear displacement. Consequently, the vector of displacement rates corresponding to DOFs in Ω, denoted $\dot{\mathbf{q}}_\Omega$, is related to the rate of the macroscopic deformation gradient $\dot{\mathbf{F}}$ by the following matrix form (more suitable for algorithmic treatment):

Table 2 Composition of sets Ψ, Ω required for the computation of the first and fourth columns of $^{m}\mathbf{C}^{(PK1)}$

Column of $^{m}\mathbf{C}^{(PK1)}$	Set Ψ	Set Ω
First column	- the DOFs of all the nodes in the interior of the RUC; - the second and the third DOF of nodes belonging to faces $x_{01} = 0$ and $\lambda_0 / 2$; - the first and the third DOF of nodes belonging to faces $x_{02} = 0$ and $\lambda_0 / 2$; - the first and the second DOF of nodes belonging to faces $x_{03} = 0$ and $\lambda_0 / 2$	- the first DOF of nodes belonging to faces $x_{01} = 0$ and $\lambda_0 / 2$; - the second DOF of nodes belonging to faces $x_{02} = 0$ and $\lambda_0 / 2$; - the third DOF of nodes belonging to faces $x_{03} = 0$ and $\lambda_0 / 2$
Fourth column	- the DOFs of all the nodes in the interior of the RUC; - the first DOF of nodes belonging to faces $x_{01} = 0$ and $\lambda_0 / 2$; - the second DOF of nodes belonging to faces $x_{02} = 0$ and $\lambda_0 / 2$; - the first and the second DOF of nodes belonging to faces $x_{03} = 0$ and $\lambda_0 / 2$	- the second and the third DOF of nodes belonging to faces $x_{01} = 0$ and $\lambda_0 / 2$; - the first and the third DOF of nodes belonging to faces $x_{02} = 0$ and $\lambda_0 / 2$; - the third DOF of nodes belonging to faces $x_{03} = 0$ and $\lambda_0 / 2$

$$\dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot {}^{v}\dot{\mathbf{F}} = \mathbf{0}, \tag{67}$$

where the topology matrix $\mathbb{S}$ is constructed in the same manner as matrix $\mathbb{D}_q$ introduced in Eq. (37), but with DOFs in set Ω.

The residual rate vector corresponding to DOFs in set Ω, denoted $\dot{\mathbf{r}}_\Omega$, can be related to the vector form $^{v}\dot{\mathbf{P}}$ of the macroscopic first Piola–Kirchhoff stress rate tensor (see Eq. (65) to understand this vector form):

$$^{v}\dot{\mathbf{P}} - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \dot{\mathbf{r}}_\Omega = \mathbf{0}. \tag{68}$$

The partition into sets Ψ and Ω is illustrated in Table 2 for the construction of the first and fourth columns of $^{m}\mathbf{C}^{(PK1)}$.

Considering this partition rule, the matrix form of the equilibrium equation associated with the RUC can be written in the following form (very similar to Eq. (43)):

$$\begin{bmatrix} \mathbf{K}_{\Psi\Psi} & \mathbf{K}_{\Psi\Omega} \\ \mathbf{K}_{\Omega\Psi} & \mathbf{K}_{\Omega\Omega} \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{q}}_\Psi \\ \dot{\mathbf{q}}_\Omega \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}}_\Psi \\ \dot{\mathbf{r}}_\Omega \end{bmatrix}. \tag{69}$$

The linearization of Eqs. (66), (67) and (68), required for finite element iterations, yields:

$$\begin{cases} \dot{\mathbf{r}}_\Psi + \mathcal{D}\dot{\mathbf{r}}_\Psi = \mathbf{0} \text{ ;} \\ \dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot {}^{v}\dot{\mathbf{F}} + \mathcal{D}\dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot \mathcal{D}({}^{v}\dot{\mathbf{F}}) = \mathbf{0} \text{ ;} \\ {}^{v}\dot{\mathbf{P}} - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \dot{\mathbf{r}}_\Omega + \mathcal{D}({}^{v}\dot{\mathbf{P}}) - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \mathcal{D}\dot{\mathbf{r}}_\Omega = \mathbf{0}. \end{cases} \tag{70}$$

Using Eq. (69), equation system (70) can be further elaborated:

$$\begin{cases} \dot{\mathbf{r}}_\Psi + \mathbf{K}_{\Psi\Psi} \cdot \mathcal{D}\dot{\mathbf{q}}_\Psi + \mathbf{K}_{\Psi\Omega} \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega = \mathbf{0} \text{ ;} \\ \dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot {}^{v}\dot{\mathbf{F}} + \mathcal{D}\dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot \mathcal{D}({}^{v}\dot{\mathbf{F}}) = \mathbf{0} \text{ ;} \\ {}^{v}\dot{\mathbf{P}} - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \dot{\mathbf{r}}_\Omega + \mathcal{D}({}^{v}\dot{\mathbf{P}}) - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \left(\mathbf{K}_{\Omega\Psi} \cdot \mathcal{D}\dot{\mathbf{q}}_\Psi + \mathbf{K}_{\Omega\Omega} \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega\right) = \mathbf{0}. \end{cases} \tag{71}$$

Equation $(71)_1$ allows us to express $\mathcal{D}\dot{\mathbf{q}}_\Psi$ as:

$$\mathcal{D}\dot{\mathbf{q}}_\Psi = -\mathbf{K}_{\Psi\Psi}^{-1} \cdot \left(\dot{\mathbf{r}}_\Psi + \mathbf{K}_{\Psi\Omega} \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega\right). \tag{72}$$

At the equilibrium state, we have $\dot{\mathbf{r}}_\Psi = \mathbf{0}$. Therefore, Eq. (72) can be reduced to the following form:

$$\mathcal{D}\dot{\mathbf{q}}_\Psi = -\mathbf{K}_{\Psi\Psi}^{-1} \cdot \mathbf{K}_{\Psi\Omega} \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega. \tag{73}$$

Insertion of Eq. (73) into Eqs. (71)$_{2,3}$ leads to the reduced equation system:

$$\begin{cases} \dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot {}^v\dot{\mathbf{F}} + \mathcal{D}\dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot \mathcal{D}\left({}^v\dot{\mathbf{F}}\right) = \mathbf{0} \; ; \\ {}^v\dot{\mathbf{P}} - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \dot{\mathbf{r}}_\Omega + \mathcal{D}\left({}^v\dot{\mathbf{P}}\right) - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \left(\mathbf{K}_{\Omega\Omega} - \mathbf{K}_{\Omega\Psi} \cdot \mathbf{K}_{\Psi\Psi}^{-1} \cdot \mathbf{K}_{\Psi\Omega}\right) \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega = \mathbf{0}. \end{cases} \tag{74}$$

Together with equation system (70), the equilibrium state of (74) requires:

$$\begin{cases} \mathcal{D}\dot{\mathbf{q}}_\Omega - \mathbb{S}^T \cdot \mathcal{D}\left({}^v\dot{\mathbf{F}}\right) = \mathbf{0} \; ; \\ \mathcal{D}\left({}^v\dot{\mathbf{P}}\right) - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \left(\mathbf{K}_{\Omega\Omega} - \mathbf{K}_{\Omega\Psi} \cdot \mathbf{K}_{\Psi\Psi}^{-1} \cdot \mathbf{K}_{\Psi\Omega}\right) \cdot \mathcal{D}\dot{\mathbf{q}}_\Omega = \mathbf{0}. \end{cases} \tag{75}$$

The elimination of $\mathcal{D}\dot{\mathbf{q}}_\Omega$ in (75) allows us to obtain:

$$\mathcal{D}\left({}^v\dot{\mathbf{P}}\right) - \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \left(\mathbf{K}_{\Omega\Omega} - \mathbf{K}_{\Omega\Psi} \cdot \mathbf{K}_{\Psi\Psi}^{-1} \cdot \mathbf{K}_{\Psi\Omega}\right) \cdot \mathbb{S}^T \cdot \mathcal{D}\left({}^v\dot{\mathbf{F}}\right) = \mathbf{0}. \tag{76}$$

The expression of the matrix form ${}^m\mathbf{C}^{(PK1)}$ of the overall tangent modulus can be easily identified from Eq. (76):

$$ {}^m\mathbf{C}^{(PK1)} = \frac{1}{|\mathcal{V}_{r0}|}\mathbb{S} \cdot \left(\mathbf{K}_{\Omega\Omega} - \mathbf{K}_{\Omega\Psi} \cdot \mathbf{K}_{\Psi\Psi}^{-1} \cdot \mathbf{K}_{\Psi\Omega}\right) \cdot \mathbb{S}^T. \tag{77}$$

6 Numerical results

6.1 Basic validations of the three techniques

To validate the implementation of the three techniques (namely, CT, FT, and PT), the numerical predictions obtained by these techniques are compared with those given by Miehe et al. [37]. Consistently, two plane composite microstructures are considered (width/length = 1/1). Each microstructure is made of a soft matrix reinforced by a stiff inclusion. The behavior of each phase is assumed to be isotropic linear elastic with the following elasticity parameters:

- Matrix: $E_m = 2081.06$ MPa; $\quad \nu_m = 0.3007$.
- Inclusion: $E_i = 10\, E_m$; $\quad \nu_i = \nu_m$.

For the first microstructure, the inclusion is a layer centered in the middle of the composite and occupying 1/3 of the total volume (Fig. 5). As to the second microstructure, the associated inclusion is a centered cylindrical fiber and its volume fraction is equal to 12.56% (Fig. 6).

To compute the macroscopic tangent moduli by the different techniques, a plane strain loading has been applied on the two analyzed microstructures (the same as the one defined in [37]). This macroscopic loading is defined by the following deformation history:

$$\text{Initial state:} \quad \mathbf{F}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{final state:} \quad \mathbf{F}(t_f) = \begin{bmatrix} 1.03 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{78}$$

with t_f corresponding to the end of the loading history, and the deformation gradient $\mathbf{F}$ being assumed to evolve linearly between $\mathbf{F}(0)$ and $\mathbf{F}(t_f)$.

In this case, we report and compare the in-plane components of the macroscopic moduli $\mathbf{C}^{(PK1)}$ stored in matrix $\tilde{\mathbf{C}}$:

$$\tilde{\mathbf{C}} = \begin{bmatrix} C_{1111}^{(PK1)} & C_{1122}^{(PK1)} & 0 \\ C_{2211}^{(PK1)} & C_{2222}^{(PK1)} & 0 \\ 0 & 0 & C_{1212}^{(PK1)} \end{bmatrix}. \tag{79}$$

The difference between the results from [37] and our predictions obtained by the three techniques is quantified by a scalar indicator m defined as:

$$m = \|\tilde{\mathbf{C}}\| / \left\|\tilde{\mathbf{C}}^{Ref}\right\| = \left(\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\tilde{C}_{ij}\right)^2\right)^{1/2} \bigg/ \left(\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\tilde{C}_{ij}^{Ref}\right)^2\right)^{1/2}, \tag{80}$$

where $\tilde{\mathbf{C}}^{Ref}$ and $\tilde{\mathbf{C}}$ denote the overall modulus determined in reference [37] and that computed by our predictions at the end of the loading history, respectively.

6.1.1 Microstructure with centered layer

To investigate the effect of mesh discretization on the prediction of the tangent modulus, this microstructure is discretized by two different meshes: Mesh 1 made of 36 finite elements, as displayed in Fig. 5a, and Mesh 2 composed of 144 finite elements, as shown in Fig. 5b.

The components of $\tilde{\mathbf{C}}$ obtained by the different techniques (namely, CT, FT, and PT) with the two meshes (Mesh 1 and Mesh 2) are reported in Table 3. For the three techniques, the predictions are almost insensitive to the mesh density, thus confirming the earlier observations made in reference [37]. Moreover, the three techniques provide the same results with indicator m very close to 1, thus implying that our predictions are very close to those given in [37].

6.1.2 Microstructure with centered cylindrical fiber

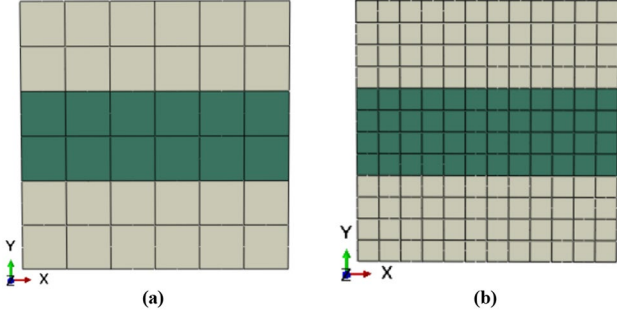In this case, the unit cell is discretized by 700 elements (Fig. 6).

Fig. 5 The finite element discretization of microstructure with centered layer: **a** Mesh 1; **b** Mesh 2

As reported in Table 4, the results obtained by CT, FT, and PT are almost identical and very close to the reference values. These results provide additional validation of our implementations.

6.2 More advanced validations of the three techniques

In the previous section, some basic validations of the three techniques have been conducted by comparing our predictions with the results presented in [37]. It appears from this preliminary study that all of the three techniques accurately predict the macroscopic tangent modulus. Thus, the
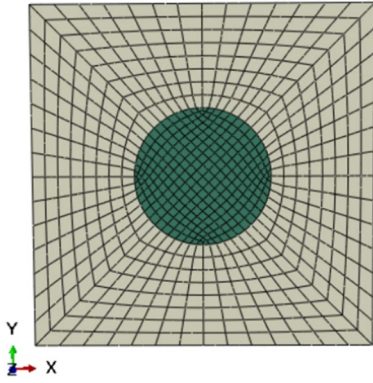

Fig. 6 The finite element discretization of microstructure with centered cylindrical fiber

accuracy and reliability of the implementation of the three techniques are partially validated. In this section, attention is focused on evaluating the computational performances of the three techniques, by considering two microstructure examples:

- Microstructure with cubic inclusion: the geometry is characterized by a cube containing a stiff cubic inclusion in the center, which occupies 20% of the total volume (Fig. 7a). This microstructure is discretized by 1000 finite elements.
- Microstructure with elliptical cylindrical fiber: the geometry is characterized by a cube containing a stiff elliptical cylindrical fiber in the center, which occupies 12.6% of the total volume (Fig. 7b). The fiber is aligned in the Z-direction and its cross section is an ellipse with aspect ratio (long axis/short axis) equal to 2. This microstructure is discretized by 2176 finite elements.

For the two cases, the mechanical behavior of the matrix and the inclusion are assumed to be elastoplastic and linear elastic, respectively:

- The matrix elasticity and hardening parameters are:

$$E_m = 210 \text{ GPa} \quad ; \quad v_m = 0.3 \quad ;$$

isotropic hardening law: $\sigma_y = 362.99 \left(0.008 + \varepsilon_{eq}^p \right)^{0.184}$.

- The inclusion elasticity parameters are:

$$E_i = 10 \, E_m \quad ; \quad v_i = v_m.$$

The two microstructures are subjected to the following deformation history:

$$\text{Initial state:} \quad \mathbf{F}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{final state:}$$

$$\mathbf{F}(t_f) = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 0.91287 & 0 \\ 0 & 0 & 0.91287 \end{bmatrix}, \tag{81}$$

Table 3 Components of matrix $\tilde{\mathbf{C}}$ for the two meshes of the microstructure with centered layer

	Reference [37]		CT		FT		PT	
	Mesh 1	Mesh 2	Mesh 1	Mesh 2	Mesh 1	Mesh 2	Mesh 1	Mesh 2
$\tilde{C}_{11}$	78,682.6	78,682.6	78,564.6	78,564.4	78,564.6	78,564.4	78,564.5	78,564.5
$\tilde{C}_{22}$	4204.0	4204.0	4189.5	4189.5	4189.5	4189.5	4189.5	4189.5
$\tilde{C}_{12}$	1815.9	1815.9	1801.5	1801.5	1801.5	1801.5	1801.5	1801.5
$\tilde{C}_{33}$	1194.0	1194.0	1194.0	1194.0	1194.0	1194.0	1194.0	1194.0
m	1.000	1.000	0.998	0.998	0.998	0.998	0.998	0.998

Table 4 Components of matrix $\tilde{\mathbf{C}}$ for the microstructure with centered cylindrical fiber

	Reference [37]	CT	FT	PT
$\tilde{C}_{11}$	3413.1	3400.7	3400.7	3400.8
$\tilde{C}_{22}$	3413.1	3400.8	3400.8	3400.8
$\tilde{C}_{12}$	1415.1	1407.2	1407.2	1407.2
$\tilde{C}_{33}$	960.1	958.8	958.8	958.9
m	1.000	0.996	0.996	0.996

(a)   **(b)**

Fig. 7 Two typical composites discretized by finite elements: **a** Microstructure with cubic inclusion; **b** microstructure with elliptical cylindrical fiber

with t_f corresponding to the end of the loading history, and the deformation gradient $\mathbf{F}$ being assumed to evolve linearly between $\mathbf{F}(0)$ and $\mathbf{F}(t_f)$.

During the general finite element computations, the external files required to compute the overall tangent moduli are automatically created and filled. Based on these external files, the developed Python scripts are executed. For CT, the required external file is the '.mtx' file, which contains the elementary stiffness matrices $\mathbf{K}_{el}$ at each n converged increments (with n being the record frequency, as shown in Appendix A). For FT, besides this '.mtx' file, two other '.txt' files are required, which contain the micro tangent moduli $\mathbf{c}^{(PK1)}$ for all the integration points and the elementary fluctuation matrices $\hat{\mathbf{K}}_{el}$. For PT, the external files are the database needed to conduct the restart analysis. This data includes the '.res', '.mdl' and '.stt', '.prt', '.odb' files, as well as the '.sim' files. Some practical aspects related to the execution of the Python scripts can be found in Sect. 4 and in the appendices. Note that the disk space allocated for the generated external files, and the CPU time spent for the tangent modulus computations increase with the complexity of the studied microstructures. Therefore, the evaluation of the computational efficiency is twofold: the required disk space and CPU time. These computations were made on eight parallelized cores allocated in cluster computer.

6.2.1 Microstructure with a cubic inclusion

The mechanical behavior of the two phases is assumed to be isotropic (for both elasticity and plasticity), and the von Mises yield function is used to compute the equivalent stress from the stress tensor. The evolution of the components $C_{1111}^{(PK1)}$, $C_{2222}^{(PK1)}$, $C_{1122}^{(PK1)}$, and $C_{1212}^{(PK1)}$ obtained by CT, FT, and PT are reported in Fig. 8. It is clear from this figure that the three techniques give identical results, thus providing additional validation of our implementation.

The overall modulus $\mathbf{C}^{(PK1)}$ is evaluated at each $\Delta t = 0.01\, t_f$ (see Eq. (81)). Then, $\mathbf{C}^{(PK1)}$ is computed 100 times during the loading history. As shown in Table 5, PT consumes more disk space and CPU time than CT and FT. This result is expectable considering the fact that ten finite element computations are required when the PT is used,
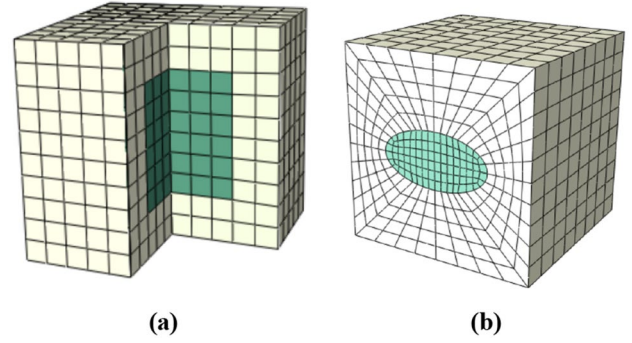
against only a single computation for the other techniques. Despite the fact that the nine finite element steps required to numerically construct the overall tangent modulus by the PT are linear (hence, relatively quick to be run), the CPU time consumed by the computations in this case remains relatively high. Even though there is no much difference between the CPU times consumed by CT and FT, the external file size required by FT is almost twice that required by CT. This result is also expectable considering the amount of data to be outputted when the FT is applied.

6.2.2 Microstructure with elliptical cylindrical fiber

In this case, the plastic behavior of the matrix is assumed to be anisotropic and it is modeled by the Hill'48 yield function with Lankford coefficients $r_0 = 0.585$; $r_{45} = 0.571$; and $r_{90} = 0.766$. The evolutions of the components $C_{1111}^{(PK1)}$, $C_{2222}^{(PK1)}$, $C_{1122}^{(PK1)}$, and $C_{1212}^{(PK1)}$ obtained by the three implemented techniques are plotted in Fig. 9. This figure confirms once again that the three techniques provide identical results.

The overall tangent modulus is evaluated at each $\Delta t = 0.04\, t_f$ (see Eq. (81)). As shown in Table 6, PT consumes much more CPU time and requires the largest disk space. Also, as previously shown, CT appears to be the most efficient in terms of CPU times and external files.

6.3 Numerical assessment of extended formulation of condensation technique in symmetric microstructures

We have presented in Sect. 5 the extended formulations of CT as well as of PT for the case of symmetric microstructures. In the same way as in the previous Sect. 6.2, the comparison between the performances of extended CT and PT has been conducted for reduced unit cells (RUC). The results of such a comparison (not shown here for brevity) reveal that the performance difference between CT and PT is similar to
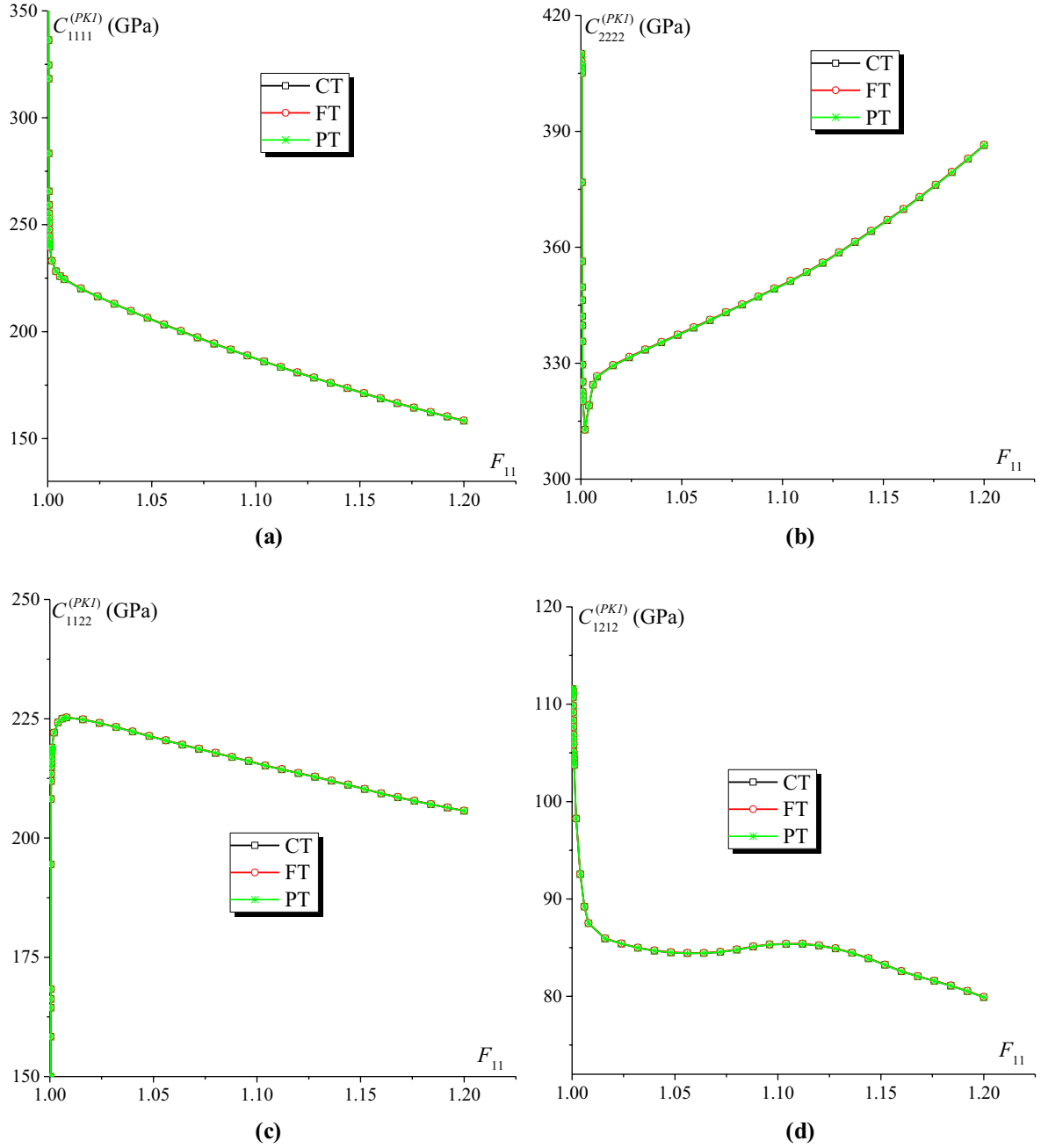
Fig. 8 Evolution of the components of modulus $\mathbf{C}^{(PK1)}$ obtained by CT, FT, and PT for microstructure with a cubic inclusion: **a** $C_{1111}^{(PK1)}$; **b** $C_{2222}^{(PK1)}$; **c** $C_{1122}^{(PK1)}$; **d** $C_{1212}^{(PK1)}$

the case of original formulations with full unit cells (FUC) (which have been reported in Tables 5 and 6). Consequently, attention is confined in this subsection to the performance analysis of the extended CT. The objective is to compare the performance of the extended formulation with RUC to the performance of the original formulation with FUC.

To assess the performance of the extension of CT to problems exhibiting symmetry properties (Sect. 5), let us consider the microstructure with elliptical cylindrical fiber.

Table 5 Performance of the different techniques in terms of allocated space disk and CPU time for microstructure with cubic inclusion

	CT	FT	PT
External files (GB)	3.979	7.654	11.087
CPU time (min)	82	85.1	111.64
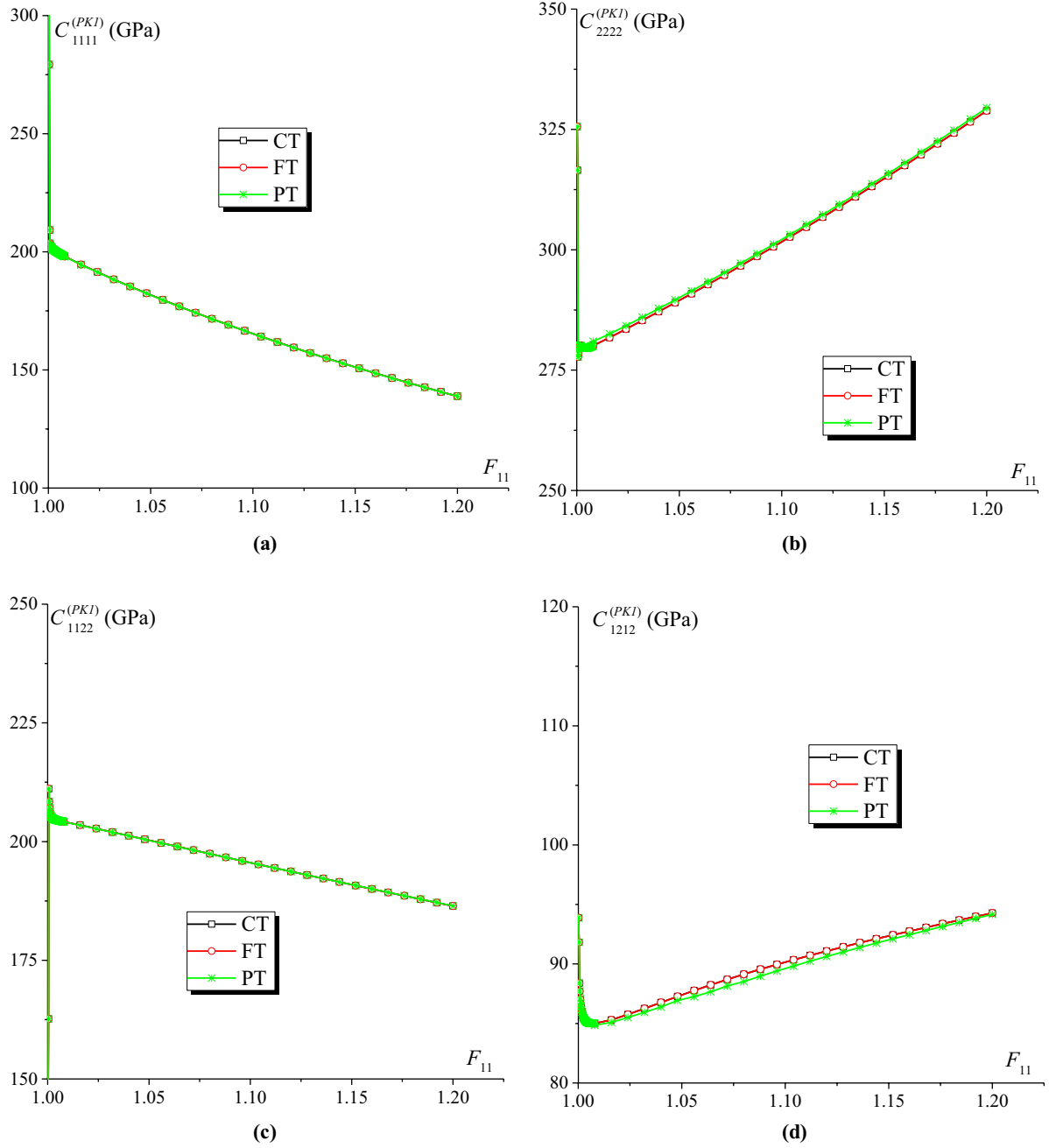
Fig. 9 Evolution of the components of modulus $\mathbf{C}^{(PK1)}$ obtained by CT, FT, and PT for microstructure with an elliptical cylindrical inclusion: **a** $C_{1111}^{(PK1)}$; **b** $C_{2222}^{(PK1)}$; **c** $C_{1122}^{(PK1)}$; **d** $C_{1212}^{(PK1)}$

Table 6 Performance of the different techniques in terms of allocated space disk and CPU time for microstructure with elliptical cylindrical fiber

	CT	FT	PT
External files (GB)	2.064	3.977	4.48
CPU time (min)	24	26.7	40.5

The RUC is discretized by 272 elements (Fig. 10) versus 2176 elements for FUC (Fig. 7b). The material parameters of both phases (matrix and inclusion) are the same as those provided in Sect. 6.2.

Both FUC and RUC are subjected to the following deformation history:
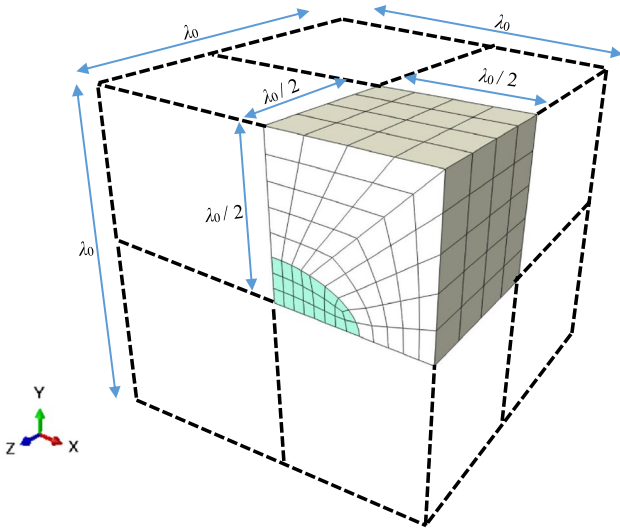
Fig. 10 RUC with elliptical cylindrical fiber

Initial state: $\mathbf{F}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow$ final state: $\mathbf{F}(t_f) = \begin{bmatrix} 1. & 0.2 & 0 \\ 0 & 1. & 0 \\ 0 & 0 & 1. \end{bmatrix}$,

$$(82)$$

which corresponds to a simple shear test.

The distribution of the von Mises equivalent stress and the maximum principal logarithmic strain obtained for both unit cells (namely FUC and RUC) at the end of the loading are displayed in Fig. 11. As clearly shown in this figure, the two unit cell configurations give very close mechanical responses. These results prove the reliability and accuracy of the boundary conditions applied on the boundary of the RUC, as summarized in Table 1.

Figure 12 provides the evolution of components $C_{1111}^{(PK1)}$, $C_{2222}^{(PK1)}$, $C_{1122}^{(PK1)}$, and $C_{1212}^{(PK1)}$ as predicted by both the original CT with FUC and the extended CT version with RUC. The perfect agreement between the predictions observed in this figure clearly demonstrates that the extended condensation technique is correctly implemented and is reliable.

The overall tangent modulus is evaluated at each $\Delta t = 0.04\, t_f$. As shown in Table 7, the use of the RUC instead of the FUC for the computation of the macroscopic tangent modulus allows dividing the size of external files by 8 and the CPU time by 13. This means that the RUC model greatly improves the computational efficiency. It is worth noting that, for CT, the CPU time is essentially consumed by the process of inverting large matrices. In the present comparative study, the element number used for the RUC model is reduced by a factor of 8, compared to the FUC model, which leads to a strong reduction in the stiffness

(a)

(b)

(c)

(d)
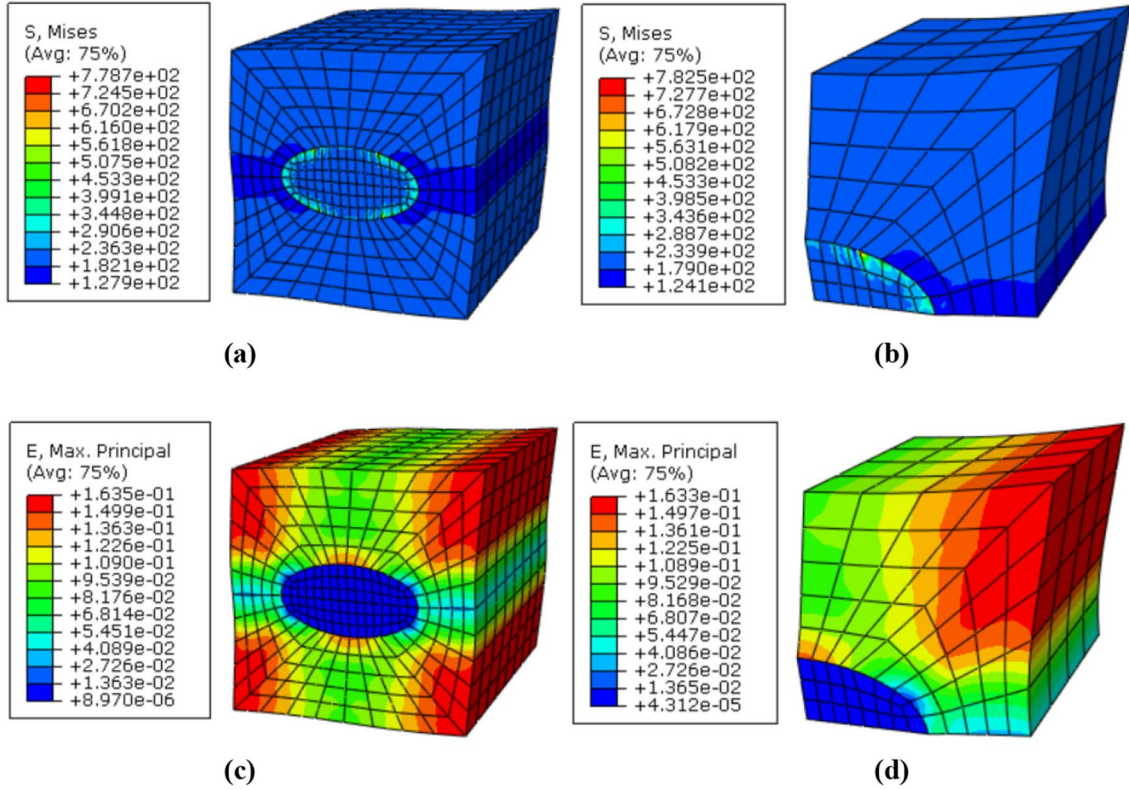
Fig. 11 Mechanical responses for both unit cells: **a** contour plot of the von Mises equivalent stress for the FUC; **b** contour plot of the von Mises equivalent stress for the RUC; **c** contour plot of the maximum principal logarithmic strain for the FUC; **d** contour plot of the maximum principal logarithmic strain for the RUC
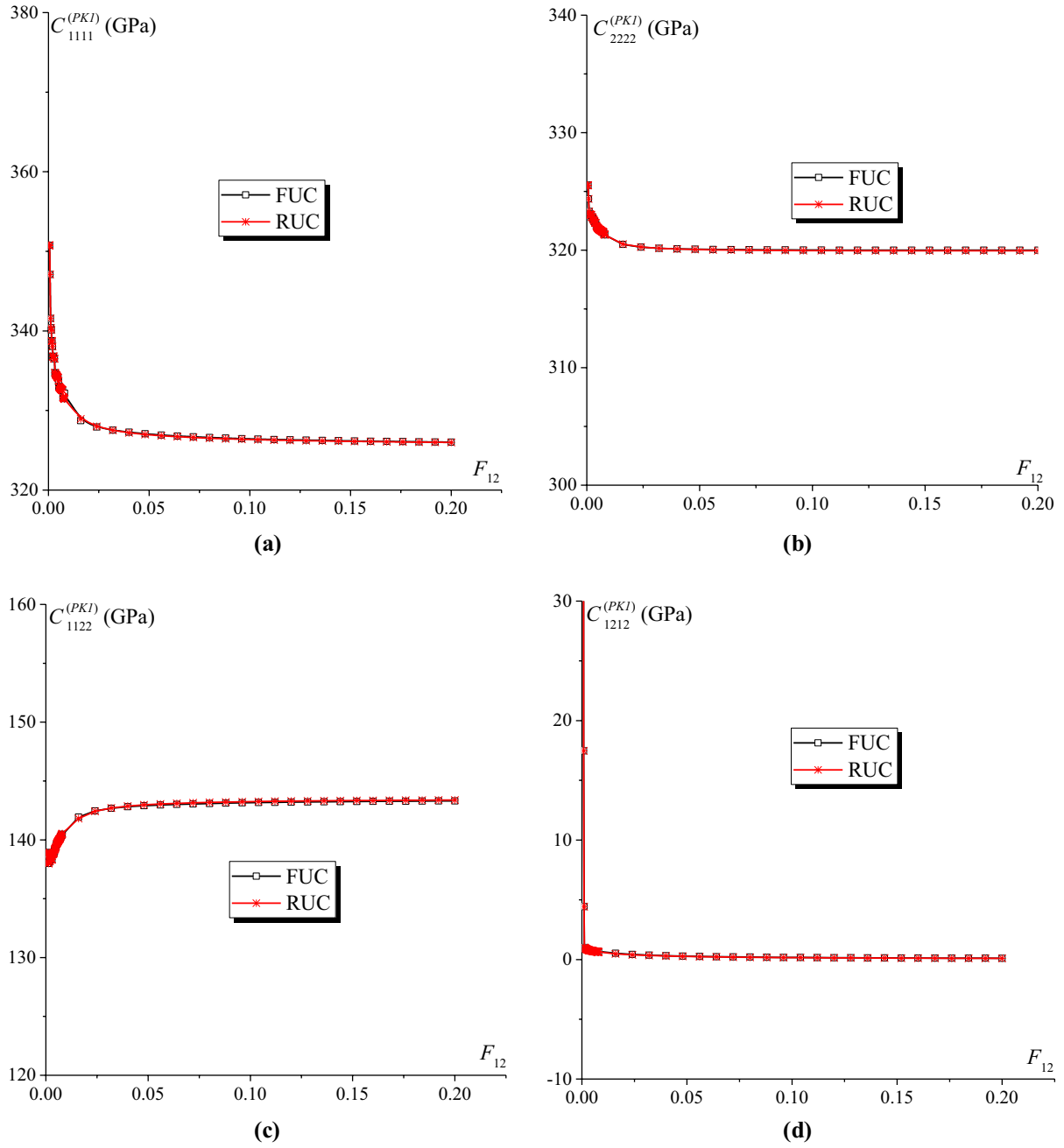
Fig. 12 Evolution of the components of modulus $\mathbf{C}^{(PK1)}$ as predicted by the original CT with FUC and the extended CT with RUC: **a** $C_{1111}^{(PK1)}$; **b** $C_{2222}^{(PK1)}$; **c** $C_{1122}^{(PK1)}$; **d** $C_{1212}^{(PK1)}$

Table 7 Disk space and CPU time required for the use of CT with FUC and RUC

	FUC	RUC
External files (GB)	2.064	0.26
CPU time (min)	24	1.8

matrix dimension (reduction by a factor of 43.84 in the current model). This strong size reduction induces a significant decrease in the computational effort. This comparison highlights the great interest of using the RUC model when the microstructure exhibits symmetry properties.

7 Summary and conclusions

In this paper, three numerical techniques used to compute the overall tangent moduli for periodic unit cells have been briefly presented and implemented within ABAQUS/Standard by developing a set of Python scripts. Several conclusions can be drawn on the basis of the study conducted to compare the different techniques:

- The perturbation technique can be carried out by using only the *Homtools* (for the application of the periodic boundary conditions and the macroscopic loading) and the Python scripts that we have developed. Hence, additional subroutines or tools (such as UMAT or UEL) are not essential for the computation of the overall tangent moduli by this technique. The numerical predictions reveal that PT is the most expensive both in terms of CPU time and external file storage. The high CPU time is attributable to the necessity to perform ten finite element computation steps (1 general step and 9 perturbation steps) for each computation of the tangent modulus.
- To implement and run the fluctuation technique, a user element (UEL) subroutine needs to be used. This limits the potential use of this method by the ABAQUS users' community. Furthermore, even though this technique does not consume excessive CPU time, the disk space occupied by the external files generated by this method remains relatively large as compared to the case of the condensation method.

- Compared to the latter techniques, the condensation technique seems to be easier to operate and reveals to be timesaving. To use this technique in conjunction with inelastic behavior, a user material (UMAT) subroutine should be used. Moreover, this technique has been successfully extended to explore microstructures exhibiting symmetry properties, and the interest of this extension has been highlighted through some numerical predictions.

Thus, this paper provides valuable reference guidelines to ABAQUS/Standard users for the determination of the homogenized tangent moduli of linear or nonlinear heterogeneous materials, such as composites, polycrystalline aggregates and porous solids. The techniques and tools developed in this paper could be used, after some adaptations, in connection with other commercial FE software packages.

Appendix A: Typical input file to output elementary stiffness matrices

In this example, the statements in red are required to be added to the input file. In this additional part, 'Nel' and 'Frequency = n' denote the total number of elements and the frequency of record of the elementary stiffness matrices (each 'n' increments).

```
…
** ASSEMBLY
**
*Assembly, name=Assembly
…
*Elset, elset=Set-1, internal, instance=Part-1-1, Generated
1, Nel, 1
*End Assembly
…
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=YES, inc=50000
…
*Element Matrix Output, Elset=Set-1, File Name=myMatrix, Frequency=n, Output File=User Defined,
Stiffness=Yes
*End Step
```

Appendix B: Some sections of Python scripts

B.1. The perturbation technique

```
#=================================================================================
# A restart file with perturbation (magnitude 10⁻⁶) for component 11 of F, restart from increment 400 of step 1
#=================================================================================
*Heading
*RESTART, READ, STEP=1, INC=400, END STEP
** ------------------------------------------------------------
**
** STEP: Step-2
**
*STEP, PERTURBATION
*Static
**
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
Set-1806, 1, 1, 1e-06
Set-1806, 2, 2
Set-1806, 3, 3
** Name: BC-2 Type: Displacement/Rotation
*Boundary
Set-1807, 1, 1
Set-1807, 2, 2
Set-1807, 3, 3
** Name: BC-3 Type: Displacement/Rotation
*Boundary
Set-1808, 1, 1
Set-1808, 2, 2
Set-1808, 3, 3
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT

*End Step
```

B.2. The condensation technique

```python
#==============================================================================
# This function is capable of computing macro moduli at increment i
#==============================================================================
def macro_tangent(labelA,i,dictionary): # labelA is an array used to store the labels of nodes in interior
    initial_Ki=np.asarray(initial_Ki)
    Kab=np.delete(Kaa_Kab,aaindex,axis=1)
    Kba=Kab.transpose()
    Kbb=np.delete(initial_Ki,aaindex,axis=0)
    Kbb=np.delete(Kbb,aaindex,axis=1)
    inv_KaaKab=np.linalg.solve(Kaa,Kab)
    bb=np.dot(Kba,inv_KaaKab)
    condense_Kbb=Kbb-bb
    inv_condense_Kbb=linalg.pinvh(condense_Kbb)
    effective_K_temp=np.dot(dot(P,inv_condense_Kbb),PT)
    effective_K=linalg.pinvh(effective_K_temp)
#==============================================================================
# Compute C
#==============================================================================
    C=(1/V)*(np.dot(dot(Q,effective_K),QT))
    dictionary[i]=[C]
#==============================================================================
num_proc=15 # specify the number of processor
manager = multiprocessing.Manager()
dictionaryP=manager.dict()
#==============================================================================
# Parallel for the calculation
#==============================================================================
for ii in range (0,incnum,num_proc):
    currentUpLimit=ii+num_proc
    if (currentUpLimit>incnum):
        currentUpLimit=incnum
    processes = [multiprocessing.Process(target= macro_tangent,args=(labelA,kk,dictionaryP))
for kk in range(ii,currentUpLimit)]
    [process.start() for process in processes]
    [process.join() for process in processes]
#==============================================================================
# Write C into a ".txt" file
#==============================================================================
fo = open("C-con.txt", "w")
for jj in range (incnum):
    L=dictionaryP[jj]
    fo.write('inc=')
    fo.write('%d\n'%incnum_sav[jj])
    for kk in range(9):
        fo.write('%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\n'\%(L[0][kk][0],L[0][kk][1],
        L[0][kk][2],L[0][kk][3],L[0][kk][4],L[0][kk][5],L[0][kk][6],L[0][kk][7],L[0][kk][8]))
fo.close()
```

B.3. The fluctuation technique

```python
#==============================================================================
# This function is capable of computing the softening part and macro moduli, at increment j
#==============================================================================
def softeningpart(j,dictionary):
    C=np.zeros((9,9))
    initial_Kj=np.asarray(initial_Kj)
    initial_Lj=np.asarray(initial_Lj)
#==============================================================================
# Compute C
#==============================================================================
    inv_initial_Kj=linalg.pinvh(initial_Kj)
    softeningpart=np.dot(dot(initial_LjT,inv_initial_Kj),initial_Lj)
    for i in range(elenum):
        C=C+C_ele[i]
    C=C/V-softeningpart/V
    dictionary[j]=[C]
#==============================================================================
num_proc=15 # specify the number of processor
manager = multiprocessing.Manager()
dictionaryP=manager.dict()
#==============================================================================
# Parallel for the calculation
#==============================================================================
for ii in range (0,incnum,num_proc):
    currentUpLimit=ii+num_proc
    if (currentUpLimit>incnum):
        currentUpLimit=incnum
    processes    =    [multiprocessing.Process(target=    softeningpart,args=(kk,dictionaryP))    for    kk    in
    range(ii,currentUpLimit)]
    [process.start() for process in processes]
    [process.join() for process in processes]
#==============================================================================
# Write C into a ".txt" file
#==============================================================================
fo = open("C-flu.txt", "w")
for jj in range (incnum):
    L=dictionaryP[jj]
    fo.write('inc=')
    fo.write('%d\n'%incnum_total[jj])
    for kk in range(9):
fo.write('%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\t%10.3f\n'\%(L[0][kk][0],L[0][kk][1],L[0][
kk][2],L[0][kk][3],L[0][kk][4],L[0][kk][5],L[0][kk][6],L[0][kk][7],L[0][kk][8]))
fo.close()
```

References

1. Hashin Z, Shtrikman S (1962) On some variational principles in anisotropic and nonhomogeneous elasticity. J Mech Phys Solids 10:335–342
2. Hill R (1963) Elastic properties of reinforced solids: some theoretical principles. J Mech Phys Solids 11:357–372
3. Ponte Castaneda P, Willis JR (1988) On the overall properties of nonlinearly viscous composites. Proc R Soc A Math Phys Eng Sci 416:217–244
4. Suquet PM (1993) Overall potentials and extremal surfaces of power law or ideally plastic composites. J Mech Phys Solids 41:981–1002
5. Teply JL, Dvorak GJ (1988) Bounds on overall instantaneous properties of elastic-plastic composites. J Mech Phys Solids 36:29–58
6. Willis JR (1981) Variational and related methods for the overall properties of composites. Adv Appl Mech 21:1–78
7. Mura T, Shodja HM, Hirose Y (1988) Inclusion problems. Appl Mech Rev 41:118–127
8. Nemat-Nasser S, Hori M (1996) Micromechanics: overall properties of heterogeneous materials. J Appl Mech 63:561
9. Ghossein E, Lévesque M (2014) A comprehensive validation of analytical homogenization models: the case of ellipsoidal particles reinforced composites. Mech Mater 75:135–150
10. Mercier S, Molinari A, Berbenni S, Berveiller M (2012) Comparison of different homogenization approaches for elasticviscoplastic materials. Model Simul Mater Sci Eng 20:1–22
11. Moulinec H, Suquet P (1994) A fast numerical method for computing the linear and nonlinear mechanical properties of composites. Comptes Rendus Académie Sci Série II Mécanique, Phys Chim Astron 318:1417–1423
12. Moulinec H, Suquet P (1995) A FFT-based numerical method for computing the mechanical properties of composites from images of their microstructures. Solid Mech Appl 37:235–246
13. Moulinec H, Suquet P (2003) Comparison of FFT-based methods for computing the response of composites with highly contrasted mechanical properties. Phys B Condens Matter 338:58–60
14. Moulinec H, Silva F (2014) Comparison of three accelerated FFT-based schemes for computing the mechanical response of composite materials. Int J Numer Methods Eng 97:960–985
15. Michel JC, Moulinec H, Suquet P (1999) Effective properties of composite materials with periodic microstructure: a computational approach. Comput Methods Appl Mech Eng 172:109–143
16. Miehe C (2002) Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. Int J Numer Methods Eng 55:1285–1322
17. Miehe C (2003) Computational micro-to-macro transitions for discretized micro-structures of heterogeneous materials at finite strains based on the minimization of averaged incremental energy. Comput Methods Appl Mech Eng 192:559–591
18. Kanit T, Forest S, Galliet I, Mounoury V, Jeulin D (2003) Determination of the size of the representative volume element for random composites: statistical and numerical approach. Int J Solids Struct 40:3647–3679
19. Ben Bettaieb M, Lemoine X, Duchêne L, Habraken AM (2011) On the numerical integration of an advanced Gurson model. Int J Numer Meth Eng 85:1049–1072
20. Feyel F (1999) Multiscale FE2 elastoviscoplastic analysis of composite structures. Comp Mater Sci 16:344–354
21. Lejeunes S, Bourgeois S (2011) Une Toolbox Abaqus pour le calcul de propriétés effectives de milieux hétérogènes. In: 10ème Colloq. Natl. en Calc. des Struct., pp 1–9
22. Omairey SL, Dunning PD, Sriramula S (2018) Development of an ABAQUS plugin tool for periodic RVE homogenisation. Eng Comput 35:567–577
23. Tchalla A, Belouettar S, Makradi A, Zahrouni H (2013) An ABAQUS toolbox for multiscale finite element computation. Compos Part B Eng 52:323–333
24. Özdemir I, Brekelmans WAM, Geers MGD (2008) FE2 computational homogenization for the thermo-mechanical analysis of heterogeneous solids. Comput Methods Appl Mech Eng 198:602–613
25. Asada T, Ohno N (2007) Fully implicit formulation of elastoplastic homogenization problem for two-scale analysis. Int J Solids Struct 44:7261–7275
26. Terada K, Kikuchi N (2001) A class of general algorithms for multi-scale analyses of heterogeneous media. Comput Methods Appl Mech Eng 190:5427–5464
27. Matsui K, Terada K, Yuge K (2004) Two-scale finite element analysis of heterogeneous solids with periodic microstructures. Comput Struct 82:593–606
28. Ladevèze P, Loiseau O, Dureisseix D (2001) A micro–macro and parallel computational strategy for highly heterogeneous structures. Int J Numer Methods Eng 52:121–138
29. Miehe C, Bayreuther CG (2007) On multiscale FE analyses of heterogeneous structures: from homogenization to multigrid solvers. Int J Numer Methods Eng 71:1135–1180
30. Tadano Y, Yoshida K, Kuroda M (2013) Plastic flow localization analysis of heterogeneous materials using homogenization-based finite element method. Int J Mech Sci 72:63–74
31. Michel JC, Lopez-Pamies O, Ponte Castañeda P, Triantafyllidis N (2007) Microscopic and macroscopic instabilities in finitely strained porous elastomers. J Mech Phys Solids 55:900–938
32. Miehe C, Schroder J, Becker M (2002) Computational homogenization analysis in finite elasticity. Material and structural instabilities on the micro- and macroscales of periodic composites and their interaction. Comput Methods Appl Mech Eng 191:4971–5005
33. Bruno D, Greco F, Lonetti P, Nevone Blasi P, Sgambitterra G (2010) An investigation on microscopic and macroscopic stability phenomena of composite solids with periodic microstructure. Int J Solids Struct 47:2806–2824
34. Temizer I, Wriggers P (2008) On the computation of the macroscopic tangent for multiscale volumetric homogenization problems. Comput Methods Appl Mech Eng 198:495–510
35. Kiran R, Khandelwal K (2014) Complex step derivative approximation for numerical evaluation of tangent moduli. Comput Struct 140:1–13
36. Miehe C, Koch A (2002) Computational micro-to-macro transitions of discretized microstructures undergoing small strains. Arch Appl Mech 72:300–317
37. Miehe C, Schröder J, Bayreuther C (2002) On the homogenization analysis of composite materials based on discretized fluctuations on the micro-structure. Acta Mech 155:1–16
38. Miehe C, Schotte J, Lambrecht M (2002) Homogenization of inelastic solid materials at finite strains based on incremental minimization principles. Application to the texture analysis of polycrystals. J Mech Phys Solids 50:2123–2167
39. Ji W, Waas AM, Bazant ZP (2013) On the importance of work-conjugacy and objective stress rates in finite deformation incremental finite element analysis. J Appl Mech 80:041024
40. Ben Bettaieb M, Debordes O, Dogui A, Duchene L (2012) Averaging properties for periodic homogenization and large deformation. Int J Multiscale Comput Eng 10:281–293
41. Debordes O (1986) Homogenization computations in the elastic or plastic range; applications to unidirectional composite and perforated sheets. Computational Mechanics Publications, Springer, Atlanta, pp 453–458

42. Miehe C, Bayreuther CG (2006) Multilevel FEM for heterogeneous structures: from homogenization to multigrid solvers. In: Lecture notes in applied and computational mechanics, pp 361–397

43. Geers MGD, Kouznetsova V, Matouš K, Yvonnet J (2017) Homogenization methods and multiscale modeling: nonlinear problems. In: Stein E, de Borst R, Hughes TJR (eds) Encyclopedia of computational mechanics, 2nd edn, vol 198, pp 1–34

44. Salahouelhadj A, Abed-Meraim F, Chalal H, Balan T (2012) Application of the continuum shell finite element SHB8PS to sheet forming simulation using an extended large strain anisotropic elastic-plastic formulation. Arch Appl Mech 82:1269–1290

45. Léné F (1984) Contribution à l'étude des materiaux composites et de leurs endommagements. PhD thesis, Universite Paris VI