Computer Methods in Applied Mechanics and Engineering 421 (2024) 116755

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Enhanced transformation field analysis for reduced-order modeling of problems with cohesive interfaces

A. Mishra [a], P. Carrara [a,*], S. Marfia [b], E. Sacco [c], L. De Lorenzis [a]

[a] Computational Mechanics Group, Institute for Mechanical Systems, Department of Mechanical and Process Engineering, ETH Zürich, Switzerland
[b] Department of Engineering, University of Roma Tre, Italy
[c] Department of Structures for Engineering and Architecture, University of Naples Federico II, Italy

ARTICLE INFO

Keywords:
Arc length
Debonding
Plasticity
Reduced-order modeling
Transformation field analysis

ABSTRACT

We propose a novel transformation field analysis (TFA) technique designed to solve problems with cohesive interfaces. The TFA approach, originally proposed for homogenization purposes, is (i) extended to account for arbitrary boundary conditions (i.e. not only those suitable for homogenization), (ii) endowed with an iterative Newton–Raphson solution scheme involving the consistent tangent matrix and (iii) provided with an arc-length strategy to follow possible snap-back branches. The extension to arbitrary boundary conditions leads to an efficient reduced-order modeling approach for solving complex non-linear problems. Numerical tests demonstrate that the approach performs well in challenging scenarios with several interfaces, and that sharp and multiple snap-back branches can be followed. The comparison with finite element results shows that the proposed technique yields fast yet accurate solutions.

1. Introduction

The numerical treatment of highly non-linear problems with standard approaches such as the finite element (FE) method may rapidly become prohibitively expensive due to the computational cost associated with the iterative solution scheme. This is often the case in optimization problems, where multiple analyses of the same system must be performed, or when localized phenomena define the global behavior, e.g. when cohesive interfaces are used to describe cracking or debonding phenomena. A possible solution are reduced-order models, whose aim is to reduce the computational cost while retaining satisfactory accuracy.

A reduced-order modeling technique proposed in the context of multiscale computations is *transformation field analysis* (TFA), first formulated by Dvorak in [1] and later improved and extended in [2–4]. TFA aims at defining the state of a point of a body as a linear combination of elastic and inelastic (non-linear) contributions, whose kernel is composed by the solution of the standard elastic localization problem and by a set of inelastic deformation modes encoding the effects of the inelastic processes. These modes are normally pre-computed and stored during a so-called *offline phase* and then used during the actual computation of the solution, namely in the *online phase* [5,6]. The inelastic modes can be defined as representative snapshots of the solution, extracted from a set of high-fidelity non-linear computations along predefined loading paths, or by considering spatially variable ansatz functions defined over a set of sub-domains [7]. The main difference between representing the inelastic modes with numerical snapshots or analytical ansatz functions lies in the numerical burden associated with the online and the offline phases. The first choice leads to a high computational cost in the offline phase and a low cost in the online phase, however particular care must be taken in selecting the representative loading paths used to obtain the snapshots, which may need to be augmented during the online phase in case

---

[*] Corresponding author.
E-mail address: pcarrara@ethz.ch (P. Carrara).

https://doi.org/10.1016/j.cma.2024.116755
Received 13 October 2023; Received in revised form 3 January 2024; Accepted 4 January 2024

of unsatisfactory accuracy. The second choice is easy to implement, the offline stage is usually inexpensive and does not require updates during the online phase, however the online phase is computationally more expensive and its accuracy depends on the adopted domain subdivision.

In the literature, TFA is adopted to homogenize the behavior of microstructures characterized by a wide variety of inelastic phenomena, such as elasto-plasticity [8–10], damage [7,11–13], visco-elasticity and visco-plasticity [8,14–16], debonding and cracking along cohesive interfaces [7,13] and crystal plasticity [17,18], among others. The original TFA [1] involves piecewise uniform inelastic modes defined over a set of sub-domains, an assumption that, in general, requires a fine subdivision to reach reasonable accuracy, thus degrading the computational efficiency [19]. Trying to alleviate this limitation, Michel and Suquet [8,20] adopt non-uniform inelastic modes based on the snapshot paradigm. This approach is termed *nonuniform TFA* (or *NTFA*); it is further improved in [21] and adapted to the FE method by Fritzen and Böhlke [22]. A different perspective is given in [11], where the snapshots are replaced by numerically approximated Green's tensors accounting for a piecewise uniform distribution of the inelastic deformations. This can be considered the first approach that includes analytical ansatz functions for inelastic modes with a piecewise uniform distribution. The introduction of non-uniform analytical ansatz functions is due to Sacco and Marfia [23], who initially study bulk plasticity and damage and later tackle debonding and cracking along cohesive interfaces [13], leading to the *eXtended NTFA* (or *X-NTFA*) approach.

A limitation of TFA is that, being formulated within the multiscale framework, it assumes that loading and boundary conditions fulfill the Hill–Mandel relations, thus it is inapplicable to general single-scale analyses. In this paper, the focus is on single-scale problems in presence of cohesive interfaces leading to localization phenomena. A difficulty often arising in these analyses is the presence of snap-back branches in the equilibrium path due to the softening nature of the interfacial constitutive law. Tracking such branches requires the adoption of an arc-length solution scheme that further intensifies the computational cost.

This work aims at extending the TFA approach proposed by Marfia and Sacco [13] to arbitrary boundary and loading conditions (i.e. not only those suitable for homogenization), hence, to single-scale analyses. Particular attention is paid to cases where debonding or cracking can be described by means of cohesive interfaces. Therefore, a robust arc-length algorithm applicable within TFA is also proposed; to this end, the approach in [13] is further modified to allow for the definition of the consistent tangent matrix. The results obtained with the proposed strategy, which we term *Enhanced Transformation Field Analysis* (E-TFA), are compared with those from standard non-linear FE analyses. We evaluate the role played by the TFA subdivision, the performance of the proposed arc-length strategy and the computational speed-up of E-TFA with respect to FE analyses for examples featuring non-linear behavior in the bulk, at the cohesive interfaces, and in both.

The paper is organized as follows: Section 2 defines the boundary value problem to be solved; Section 3 extends the multiscale TFA paradigm to arbitrary boundary conditions; the proposed arc-length strategy based on the consistent tangent matrix is illustrated in Section 4; Section 5 presents the numerical examples and the comparison between E-TFA and FE results; and the main conclusions are summarized in Section 6.

In the following, $\dot{(\bullet)}$, $\nabla^s(\bullet)$, $\mathrm{div}(\bullet)$, $\mathrm{tr}(\bullet)$, $\mathrm{diag}(\bullet)$, $\|\bullet\|$, $(\bullet)^T$, $\langle\bullet\rangle_+$ and $\langle\bullet\rangle_-$ stand for the time derivative, the symmetric gradient, the divergence, the trace, the diagonal, the $L2$ norm, the transpose, the positive and negative part of $(\bullet)$, respectively. Also, $\boldsymbol{I}$ is the second-order identity tensor, matrix multiplication is denoted as $\boldsymbol{AB}$, while double contraction between second-order tensors or matrices, inner (dot) product and outer product between vectors are indicated respectively as $\boldsymbol{A} : \boldsymbol{B} = \sum_i \sum_j A_{ij} B_{ij}$, $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b}$ and $\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T$. We also denote with $\boldsymbol{A} \cdot \boldsymbol{b}$ the vector $\boldsymbol{t}$ with components $t_i = \sum_j A_{ij} b_j$. Finally, we denote with $[\![(\bullet)]\!]$ the jump of $(\bullet)$, and $h(\bullet)$ stands for the Heaviside function, i.e. $h(\bullet) = 1$ when $(\bullet) \geq 0$ and zero otherwise.

2. Problem statement and governing equations

This section states the problem to be solved along with its governing equations. We take as paradigmatic example a body where cohesive interfaces are embedded in an elasto-plastic bulk.

2.1. Problem statement

Within the framework of isothermal and quasi-static mechanical processes, linearized kinematics and 2D (plane stress or plane strain) conditions, we consider a domain Ω with points parameterized by the coordinate $\boldsymbol{x}$. The displacement field $\boldsymbol{u}(\boldsymbol{x})$ is induced by the imposed tractions $\bar{\boldsymbol{f}}(\boldsymbol{x}) = \lambda \bar{\boldsymbol{f}}^*(\boldsymbol{x})$ and displacements $\bar{\boldsymbol{u}}(\boldsymbol{x}) = \lambda \bar{\boldsymbol{u}}^*(\boldsymbol{x})$, applied respectively to the Neumann $\partial\Omega_N$ and Dirichlet $\partial\Omega_D$ parts of the outer boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\partial\Omega_D \cap \partial\Omega_N = \emptyset$ (Fig. 1). Also, $\boldsymbol{n}(\boldsymbol{x})$ is the outward unit normal to the boundary, $\bar{\boldsymbol{u}}^*(\boldsymbol{x})$ and $\bar{\boldsymbol{f}}^*(\boldsymbol{x})$ are reference loads, λ is a load multiplier and the total strain and stress fields are denoted with $\boldsymbol{\varepsilon}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$, respectively.

Within the bulk of the domain, there exist cohesive interfaces whose positions are a priori known and that constitute a subset $\mathcal{I} \subset \Omega$. The interfaces are assumed to have zero thickness, thus all interfacial quantities are referred to the unique interface line in the undeformed state, that is parameterized by a curvilinear coordinate ξ; we also define the mapping $g\ :\ \Omega \rightarrow \mathcal{I},\ g(\boldsymbol{x}) = \xi$. The interfacial vectorial quantities are expressed in a local orthonormal system given by the normal $\boldsymbol{w}(\xi)$ and tangential $\boldsymbol{o}(\xi)$ unit vectors. The corresponding components of a vector $(\bullet)$ are indicated with the subscripts $(\cdot)_N$ and $(\cdot)_T$, respectively (Fig. 1).

The governing equations for the bulk and the interfaces are summarized in Table 1. Note that, while Fig. 1 represents for simplicity a homogeneous domain with one interface, we will consider more complex situations with heterogeneous domains (e.g. matrices containing inclusions with different material properties) and multiple interfaces, some of which embedded within the matrix, and some others separating matrix and inclusions, possibly with different interface properties. However, for the sake of notational simplicity we omit the coordinate dependency of bulk and interface properties in Table 1.

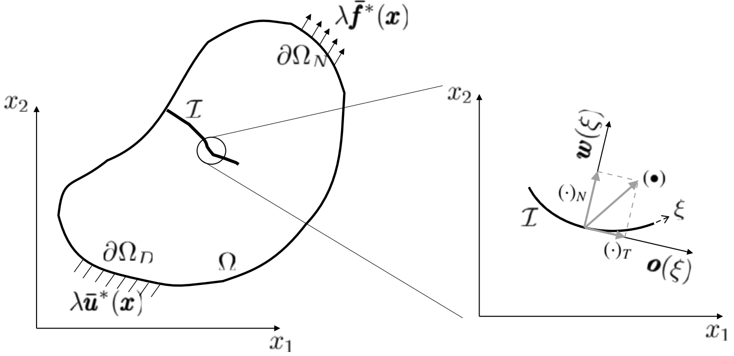


Fig. 1. Scheme of the problem.

2.2. Constitutive laws

The constitutive behavior of the bulk (Table 1) is based on an elasto-plastic law featuring linear isotropic hardening and an additive decomposition of the total strain tensor into elastic $\boldsymbol{\varepsilon}_e(\boldsymbol{x})$ and plastic $\boldsymbol{\pi}(\boldsymbol{x})$ components. The evolution of the plastic strain is governed by an associative flow rule with von Mises yield function $\varphi\left(\boldsymbol{\sigma}_d(\boldsymbol{x}), \sigma_y(\boldsymbol{x})\right)$, where $\boldsymbol{\sigma}_d(\boldsymbol{x})$ is the deviatoric stress tensor and $\sigma_y(\boldsymbol{x}) = \sigma_y(\sigma_{y0}, \alpha(\boldsymbol{x}))$ is the current yield stress, which depends on the plastic history variable (namely, the accumulated plastic strain) $\alpha(\boldsymbol{x})$ and the initial yield stress σ_{y0}. Also, H_p is the isotropic hardening modulus, $\dot{\zeta}(\boldsymbol{x})$ is the plastic multiplier and $\mathbb{C}$ is the fourth-order elasticity tensor.

Concerning the interfaces, $\boldsymbol{T}(\boldsymbol{x})$ is a projection matrix from the global to the interface local (normal-tangential) reference system, and $\boldsymbol{t}(\xi)$ is the interface traction vector. The total relative displacement $\boldsymbol{s}(\xi)$ at the interface is additively decomposed into an elastic part $\boldsymbol{s}_e(\xi)$ and an inelastic contribution $\boldsymbol{q}(\xi)$. The latter governs the non-linear behavior of the interface, that includes the effects of damage, friction and unilateral contact in the normal direction.

The interfacial constitutive behavior involves a coupled normal-tangential cohesive law following the model proposed in [24] and summarized in Table 1. The inelastic parameter $c(\xi)$ accounts for the unilateral contact, which is regularized by means of a penalty stiffness. The inelastic contribution $p(\xi)$ accounts for the interfacial friction, whose non-associative evolution is governed by a Kuhn–Tucker set of conditions involving the Coulomb friction limit function $\Phi\,(\boldsymbol{t}(\xi))$ with friction coefficient μ and inelastic multiplier $\dot{\gamma}(\xi)$. The scalar damage parameter $D(\xi)$ governs the interface decohesion following an evolution law based on the equivalent relative displacement $s_{eq}(\xi)$ and the tangential-normal coupling parameter $\eta(\xi)$. Damage, friction and unilateral contact collectively determine the inelastic relative displacement $\boldsymbol{q}(\xi)$.

The interface stiffness tensor reads

$$\boldsymbol{\kappa} = \mathrm{diag}\left(k_T, k_N\right), \tag{1}$$

where no dilatancy is accounted for and k_T and k_N are the stiffness parameters in tangential and normal direction, respectively (Fig. 2). Under monotonic tangential (i.e., mode-II) loading, we adopt a bilinear cohesive law characterized by a fracture energy $\bar{G}_{c,T}$ and a peak strength $\bar{t}_T$ corresponding to the relative displacement $\bar{s}_T$ (Fig. 2(a)). Similarly, under normal opening (i.e., mode-I) loading the bilinear envelope is characterized by the fracture energy $\bar{G}_{c,N}$, the peak strength $\bar{t}_N$ attained at a relative displacement $\bar{s}_N$, while in compression the regularized unilateral contact condition delivers a linear elastic behavior with stiffness k_N (Fig. 2(b)).

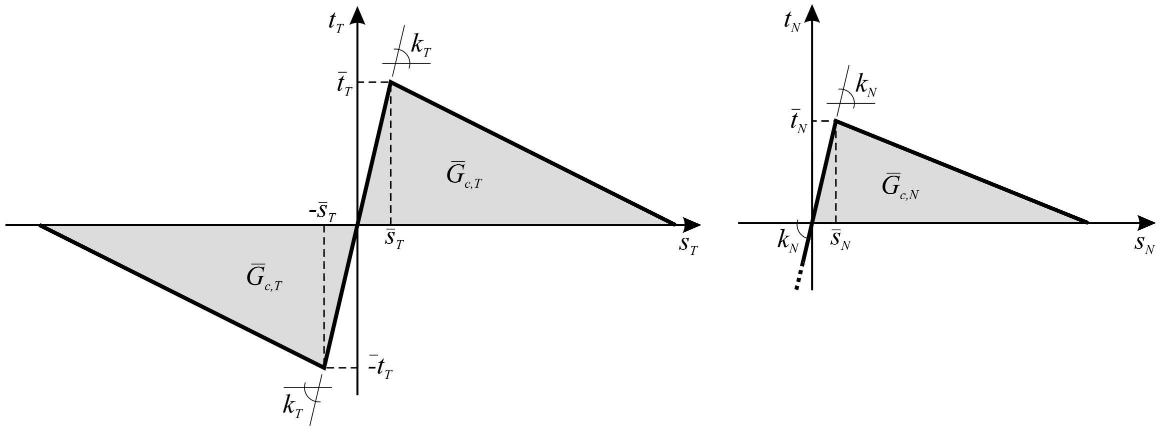
3. TFA-based solution for arbitrary boundary conditions

In this section, we first briefly summarize the standard TFA in the context of multiscale analysis, and then extend it to arbitrary boundary conditions to make it applicable to single-scale problems.

3.1. Starting point: TFA multiscale approach

In the context of a typical multiscale approach solved with TFA, the displacement gradient at a point of the macroscale is applied as average strain to a representative volume element (RVE) of the microstructure under boundary conditions satisfying the Hill–Mandel requirements. The solution of the resulting microstructural boundary value problem gives the average stress and tangent constitutive tensor at that macroscopic point.

The TFA method decomposes the microstructural problem into two sub-problems. First, the total average strain is applied to the RVE neglecting any inelastic material behavior. The second sub-problem computes the effects of the inelastic strains under the constraint of vanishing total average strain. Hence, the inelastic strain distribution is here interpreted as an eigenstrain affecting only the total average stress and is generally termed *transformation field*. The solution of the first sub-problem can be written in



(a) (b)

Fig. 2. Monotonic interfacial cohesive behavior under (a) tangential (i.e., mode-II) and (b) normal (i.e., mode-I or contact) loading.

Table 1
Governing equations of the problem in Fig. 1.

Governing equations	
Equilibrium equation	$\mathrm{div}(\boldsymbol{\sigma}(\boldsymbol{x})) = \mathbf{0}, \quad \text{in } \Omega \setminus \mathcal{I}\,,$
Boundary conditions	$\boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = \lambda\, \bar{\boldsymbol{f}}^{*}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_N\,,$ $\boldsymbol{u}(\boldsymbol{x}) = \lambda\, \bar{\boldsymbol{u}}^{*}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_D\,.$
Bulk	
Kinematics	$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \nabla^s \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\varepsilon}_e(\boldsymbol{x}) + \boldsymbol{\pi}(\boldsymbol{x}), \quad \text{in } \Omega \setminus \mathcal{I}\,,$
Constitutive law	$\boldsymbol{\sigma}(\boldsymbol{x}) = \mathbb{C} : \boldsymbol{\varepsilon}_e(\boldsymbol{x}) = \mathbb{C} : (\boldsymbol{\varepsilon}(\boldsymbol{x}) - \boldsymbol{\pi}(\boldsymbol{x})) \quad \text{in } \Omega \setminus \mathcal{I}\,,$
Evolution equations	$\varphi\left(\boldsymbol{\sigma}_d(\boldsymbol{x}), \sigma_y(\boldsymbol{x})\right) = \sqrt{\frac{3}{2}}\, \lVert\boldsymbol{\sigma}_d(\boldsymbol{x})\rVert - \sigma_y(\boldsymbol{x}) \quad \text{(Yield function)}\,,$ $\boldsymbol{\sigma}_d(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{x}) - \frac{1}{3}\,\mathrm{tr}\,(\boldsymbol{\sigma}(\boldsymbol{x}))\,\boldsymbol{I}, \quad \sigma_y(\boldsymbol{x}) = \sigma_{y0} + H_p \alpha(\boldsymbol{x}), \quad \dot{\alpha}(\boldsymbol{x}) = \dot{\zeta}(\boldsymbol{x}),$ $\dot{\boldsymbol{\pi}}(\boldsymbol{x}) = \dot{\zeta}(\boldsymbol{x}) \frac{\partial \varphi}{\partial \lVert\boldsymbol{\sigma}_d\rVert}(\boldsymbol{x}) = \dot{\zeta}(\boldsymbol{x}) \frac{\boldsymbol{\sigma}_d}{\lVert\boldsymbol{\sigma}_d\rVert}(\boldsymbol{x}) \quad \text{(Associative flow rule)}\,,$ $\dot{\zeta}(\boldsymbol{x}) \geq 0,\ \varphi\left(\boldsymbol{\sigma}_d(\boldsymbol{x}), \sigma_y(\boldsymbol{x})\right) \leq 0,\ \dot{\zeta}(\boldsymbol{x})\varphi\left(\boldsymbol{\sigma}_d(\boldsymbol{x}), \sigma_y(\boldsymbol{x})\right) = 0 \quad \text{(Kuhn–Tucker conds.)}\,,$ $\dot{\zeta}(\boldsymbol{x})\dot{\varphi}\left(\boldsymbol{\sigma}_d(\boldsymbol{x}), \sigma_y(\boldsymbol{x})\right) = 0 \quad \text{(Consistency cond.)}\,.$
Interface	
Kinematics	$\boldsymbol{s}(\xi) = \boldsymbol{T}\left(g^{-1}(\xi)\right) [\![\boldsymbol{u}\left(g^{-1}(\xi)\right)]\!] \quad \text{on } \mathcal{I}\,,$
Equilibrium with bulk	$\boldsymbol{t}(\xi) = \boldsymbol{\sigma}(g^{-1}(\xi)) \cdot \boldsymbol{w}\left(g^{-1}(\xi)\right) \quad \text{on } \Omega \cap \mathcal{I}\,,$
Constitutive law	$\boldsymbol{t}(\xi) = \boldsymbol{\kappa} \cdot \boldsymbol{s}_e(\xi) = \boldsymbol{\kappa} \cdot [\boldsymbol{s}(\xi) - \boldsymbol{q}(\xi)]\,, \quad \boldsymbol{q}(\xi) = D(\xi) \begin{vmatrix} p(\xi) \\ c(\xi) \end{vmatrix} = \begin{vmatrix} q_T(\xi) \\ q_N(\xi) \end{vmatrix}, \quad \text{on } \mathcal{I}\,,$
Evolution equations	$c(\xi) = h(s_N(\xi))s_N(\xi)\,, \quad \text{(Unilateral contact)}$ $\left.\begin{array}{r} \dot{p}(\xi) = \dot{\gamma}(\xi) \dfrac{\partial \boldsymbol{\Phi}(\boldsymbol{t}(\xi))}{\partial t_T(\xi)}\,, \quad \boldsymbol{\Phi}(\boldsymbol{t}(\xi)) = \lvert t_T(\xi)\rvert + \mu\langle t_N(\xi)\rangle_-\,, \\ \dot{\gamma}(\xi) \geq 0,\ \boldsymbol{\Phi}(\boldsymbol{t}(\xi)) \leq 0,\ \dot{\gamma}(\xi)\boldsymbol{\Phi}(\boldsymbol{t}(\xi)) = 0 \text{ and } \dot{\gamma}(\xi)\dot{\boldsymbol{\Phi}}(\boldsymbol{t}(\xi)) = 0 \end{array}\right] \text{(Friction)}$ $\left.\begin{array}{c} D(\xi) = \max\limits_{hist.}\{\,0,\, \min[1,\ D^*(\xi)]\} \\ D^*(\xi) = \dfrac{s_{eq}(\xi) - 1}{s_{eq}(\xi)(1 - \eta(\xi))}\,, \quad s_{eq}(\xi) = \sqrt{\left(\dfrac{s_T(\xi)}{\bar{s}_T}\right)^2 + \left(\dfrac{\langle s_N(\xi)\rangle_+}{\bar{s}_N}\right)^2} \\ \eta(\xi) = \dfrac{1}{\sqrt{s_T(\xi)^2 + \langle s_N(\xi)\rangle_+^2}} \left(\dfrac{\bar{s}_T \bar{t}_T}{2\bar{G}_{c,T}}\, s_T(\xi)^2 + \dfrac{\bar{s}_N \bar{t}_N}{2\bar{G}_{c,N}}\, s_N(\xi)^2\right) \end{array}\right] \text{(Damage)}$

terms of the standard elastic localization tensor from linear elastic homogenization theory. Conversely, a set of modified Green's functions able to recover the total strain at any point induced by an inelastic strain imposed at another point can be adopted to solve the inelastic eigenproblem [7,11,23,25]. Owing to the assumption of small deformation and to the linearity of the constitutive laws separately in the total and inelastic kinematic variables, the solution of the microstructural problem is obtained superposing the contributions given by the two sub-problems, which, being kinematically and statically admissible, ensure the fulfillment of the kinematic and equilibrium relationships.

Apart for being limited to multiscale settings, the application of the described procedure involves two major difficulties, namely that the inelastic eigenproblem requires the a priori knowledge of the inelastic strain distribution, and that the definition of the elastic localization and modified Green's tensors is, usually, analytically unfeasible. To overcome such difficulties, the NTFA approach proposed in [11,23] and its extension in [12,13] decompose the domain in sub-domains and sub-interfaces, while the inelastic transformation fields are parameterized and approximated within each of them using a linear combination of known spatially variable inelastic modes. Also, the localization and Green's operators are obtained numerically in a preliminary offline phase using the FE method.

3.2. Extension to arbitrary boundary conditions

The TFA approach proposed for homogenizing the behavior of an RVE is extended now to arbitrary boundary conditions. First, we reformulate the two sub-problems leading to the solution of the problem; then, we discuss the parameterization of the inelastic fields along with the numerical approximation of the localization and Green's operators.

3.2.1. TFA sub-problems

In the spirit of the described TFA analysis, the problem in Section 2 is decomposed into the following two sub-problems

- `Elastic localization problem`: where the inelastic behavior is neglected and the domain is considered purely linear elastic, namely

$$\boldsymbol{\pi}(\boldsymbol{x}) = \boldsymbol{0}\,, \quad \forall \boldsymbol{x} \in \Omega \setminus \mathcal{I}\,, \tag{2a}$$

$$\boldsymbol{q}(\xi) = \boldsymbol{0} \quad \forall \xi \in \mathcal{I}\,, \tag{2b}$$

and the boundary conditions are applied as in the original problem, i.e.

$$\boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = \lambda\, \bar{\boldsymbol{f}}^*(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_N\,, \tag{3a}$$

$$\boldsymbol{u}(\boldsymbol{x}) = \lambda\, \bar{\boldsymbol{u}}^*(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \partial\Omega_D\,. \tag{3b}$$

Owing to the linearity of the problem, the solution of the elastic localization problem is written as

$$\check{\boldsymbol{u}}(\boldsymbol{x}) = \lambda\, \boldsymbol{u}^*(\boldsymbol{x})\,,\ \forall \boldsymbol{x} \in \Omega \setminus \mathcal{I}\,, \tag{4a}$$

$$\check{\boldsymbol{\varepsilon}}(\boldsymbol{x}) = \nabla^s \check{\boldsymbol{u}}(\boldsymbol{x}) = \lambda\, \boldsymbol{\varepsilon}^*(\boldsymbol{x}) = \lambda\, \boldsymbol{\varepsilon}^*_e(\boldsymbol{x})\,, \quad \text{and} \quad \check{\boldsymbol{\sigma}}(\boldsymbol{x}) = \mathbb{C}(\boldsymbol{x}) : \check{\boldsymbol{\varepsilon}}(\boldsymbol{x})\,,\ \forall \boldsymbol{x} \in \Omega \setminus \mathcal{I}\,, \tag{4b}$$

$$\check{\boldsymbol{s}}(\xi) = \lambda\, \boldsymbol{s}^*(\xi) = \boldsymbol{T}\left(g^{-1}(\xi)\right) [\![\check{\boldsymbol{u}}\left(g^{-1}(\xi)\right)]\!]\,, \quad \text{and} \quad \check{\boldsymbol{t}}(\xi) = \boldsymbol{\kappa}(\xi) \cdot \check{\boldsymbol{s}}(\xi)\,,\ \forall \xi \in \mathcal{I}\,, \tag{4c}$$

where $\boldsymbol{u}^*(\boldsymbol{x})$, $\boldsymbol{\varepsilon}^*(\boldsymbol{x}) = \boldsymbol{\varepsilon}^*_e(\boldsymbol{x}) = \nabla^s \boldsymbol{u}^*(\boldsymbol{x})$ and $\boldsymbol{s}^*(\xi)$ are the elastic localization tensors respectively for the displacements, strains and interface displacement jumps, namely they are the solutions obtained assuming elastic material behavior and imposing the reference loads $\bar{\boldsymbol{u}}^*(\boldsymbol{x})$ on $\partial\Omega_D$ and $\bar{\boldsymbol{f}}^*(\boldsymbol{x})$ on $\partial\Omega_N$.
- `Inelastic eigenproblem`: where only the inelastic deformations are accounted for while considering homogeneous Dirichlet and Neumann boundary conditions. In this case the solution reads

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \mathbb{G}_{ub}(\boldsymbol{x}, \boldsymbol{x}') * \boldsymbol{\pi}(\boldsymbol{x}') + \mathbb{G}_{ui}(\boldsymbol{x}, \xi') * \boldsymbol{q}(\xi')\,,\ \forall \boldsymbol{x} \in \Omega \setminus \mathcal{I}\,, \tag{5a}$$

$$\tilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}) = \mathbb{G}_{bb}(\boldsymbol{x}, \boldsymbol{x}') * \boldsymbol{\pi}(\boldsymbol{x}') + \mathbb{G}_{bi}(\boldsymbol{x}, \xi') * \boldsymbol{q}(\xi')\,, \quad \text{and} \quad \tilde{\boldsymbol{\sigma}}(\boldsymbol{x}) = \mathbb{C}(\boldsymbol{x}) : (\tilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}) - \boldsymbol{\pi}(\boldsymbol{x}))\,,\ \forall \boldsymbol{x} \in \Omega \setminus \mathcal{I}\,, \tag{5b}$$

$$\begin{aligned} \tilde{\boldsymbol{s}}(\xi) = \boldsymbol{T}\left(g^{-1}(\xi)\right) [\![\tilde{\boldsymbol{u}}\left(g^{-1}(\xi)\right)]\!] = \mathbb{G}_{ib}(\xi, \boldsymbol{x}') * \boldsymbol{\pi}(\boldsymbol{x}') + \mathbb{G}_{ii}(\xi, \xi')\, * \boldsymbol{q}(\xi')\,, \quad \text{and} \\ \tilde{\boldsymbol{t}}(\xi) = \boldsymbol{\kappa}(\xi) \cdot (\tilde{\boldsymbol{s}}(\xi) - \boldsymbol{q}(\xi))\,,\ \forall \xi \in \mathcal{I}\,, \end{aligned} \tag{5c}$$

where $\mathbb{G}_{ub}(\boldsymbol{x}, \boldsymbol{x}')$ and $\mathbb{G}_{ui}(\boldsymbol{x}, \xi')$ are the Green's tensors able to recover the displacement at a point $\boldsymbol{x}$ due to an imposed strain in $\boldsymbol{x}'$ and an imposed relative displacement in ξ', respectively. Also, $\mathbb{G}_{bb}(\boldsymbol{x}, \boldsymbol{x}')$ and $\mathbb{G}_{bi}(\boldsymbol{x}, \xi')$ ($\mathbb{G}_{ib}(\xi, \boldsymbol{x}')$ and $\mathbb{G}_{ii}(\xi, \xi')$) encode the strain (resp. relative displacement) at a point $\boldsymbol{x}$ (resp. ξ) due to an imposed strain in $\boldsymbol{x}'$ and an imposed relative displacement in ξ', respectively. In (5) the operator $*$ stands for a convolution integral, i.e.

$$\mathbb{G}_{ub}(\boldsymbol{x}, \boldsymbol{x}') * \boldsymbol{\pi}(\boldsymbol{x}') = \int_{\Omega \setminus \mathcal{I}} \mathbb{G}_{ub}(\boldsymbol{x}, \boldsymbol{x}') \boldsymbol{\pi}(\boldsymbol{x}')\, d\boldsymbol{x}'\,, \quad \text{or} \quad \mathbb{G}_{ui}(\boldsymbol{x}, \xi') * \boldsymbol{q}(\xi') = \int_{\mathcal{I}} \mathbb{G}_{ui}(\boldsymbol{x}, \xi') \boldsymbol{q}(\xi')\, d\xi'\,, \tag{6}$$

and similarly for $\mathbb{G}_{bb}(\boldsymbol{x}, \boldsymbol{x}')$, $\mathbb{G}_{bi}(\boldsymbol{x}, \xi')$, $\mathbb{G}_{ib}(\xi, \boldsymbol{x}')$ and $\mathbb{G}_{ii}(\xi, \xi')$.

The inelastic strain $\boldsymbol{\pi}(\boldsymbol{x})$ and relative displacements $\boldsymbol{q}(\xi)$ are thus interpreted as eigendeformations [2,26] and collectively represent the *transformation fields*. Finally, the superposition of (4)–(5) delivers the solution of the problem in Section 2. This solution depends on the localization and Green's tensors along with bulk and interface inelastic fields, whose handling is described in the following sections.

3.2.2. Inelastic field parameterization

Inspired by [12,13], we parameterize the inelastic transformation fields using linear combinations of known spatially variable *inelastic deformation modes*. This parameterization relies on the decomposition of the domain Ω into D sub-domains and I sub-interfaces; on each of them we approximate the two fields $\boldsymbol{\pi}(\boldsymbol{x})$ and $\boldsymbol{q}(\xi)$ as

$$\boldsymbol{\pi}(\boldsymbol{x}) \simeq \sum_{d=1}^{D} \sum_{m=1}^{M_{\Omega}} \boldsymbol{B}_{d,m}(\boldsymbol{x}) \hat{\boldsymbol{\pi}}_{d,m} \,, \tag{7a}$$

$$\boldsymbol{q}(\xi) \simeq \sum_{i=1}^{I} \sum_{m=1}^{M_{\mathcal{I}}} \boldsymbol{A}_{i,m}(\xi) \hat{\boldsymbol{q}}_{i,m} \,, \tag{7b}$$

Here we are using Voigt notation; $\boldsymbol{B}_{d,m}(\boldsymbol{x})$ with $m = 1, \dots, M_{\Omega}$ are M_{Ω} matrices storing a set of predefined spatially varying inelastic deformation modes associated with the dth sub-domain Ω_d and vanishing for any $\boldsymbol{x} \notin \Omega_d$, while $\hat{\boldsymbol{\pi}}_{d,m}$ are the combination coefficients. Similarly, the parameterization for the ith sub-interface consists of $M_{\mathcal{I}}$ inelastic deformation mode matrices $\boldsymbol{A}_{i,m}(\xi)$ with $m = 1, \dots, M_{\mathcal{I}}$, vanishing for any $\xi \notin \mathcal{I}_i$, and a corresponding set of combination coefficients $\hat{\boldsymbol{q}}_{i,m}$.

Substituting (7) in (5) and exploiting the linearity of (6) we obtain

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) \simeq \sum_{d=1}^{D} \sum_{m=1}^{M_{\Omega}} \mathbb{U}_{d,m}(\boldsymbol{x}) \hat{\boldsymbol{\pi}}_{d,m} + \sum_{i=1}^{I} \sum_{m=1}^{M_{\mathcal{I}}} \mathbb{T}_{i,m}(\boldsymbol{x}) \hat{\boldsymbol{q}}_{i,m} \,, \tag{8a}$$

$$\tilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}) \simeq \sum_{d=1}^{D} \sum_{m=1}^{M_{\Omega}} \mathbb{P}_{d,m}(\boldsymbol{x}) \hat{\boldsymbol{\pi}}_{d,m} + \sum_{i=1}^{I} \sum_{m=1}^{M_{\mathcal{I}}} \mathbb{S}_{i,m}(\boldsymbol{x}) \hat{\boldsymbol{q}}_{i,m} \,, \tag{8b}$$

$$\tilde{\boldsymbol{s}}(\xi) \simeq \sum_{d=1}^{D} \sum_{m=1}^{M_{\Omega}} \mathbb{R}_{d,m}(\xi) \hat{\boldsymbol{\pi}}_{d,m} + \sum_{i=1}^{I} \sum_{m=1}^{M_{\mathcal{I}}} \mathbb{Q}_{i,m}(\xi) \hat{\boldsymbol{q}}_{i,m} \,, \tag{8c}$$

where we define the *interaction tensors*

$$\mathbb{U}_{d,m}(\boldsymbol{x}) = \mathbb{G}_{ub}(\boldsymbol{x}, \, \boldsymbol{x}') * \boldsymbol{B}_{d,m}(\boldsymbol{x}'), \tag{9a}$$

$$\mathbb{T}_{i,m}(\boldsymbol{x}) = \mathbb{G}_{ui}(\boldsymbol{x}, \, \xi') * \boldsymbol{A}_{i,m}(\xi'), \tag{9b}$$

$$\mathbb{P}_{d,m}(\boldsymbol{x}) = \mathbb{G}_{bb}(\boldsymbol{x}, \, \boldsymbol{x}') * \boldsymbol{B}_{d,m}(\boldsymbol{x}'), \tag{9c}$$

$$\mathbb{S}_{i,m}(\boldsymbol{x}) = \mathbb{G}_{bi}(\boldsymbol{x}, \, \xi') * \boldsymbol{A}_{i,m}(\xi'), \tag{9d}$$

$$\mathbb{R}_{d,m}(\xi) = \mathbb{G}_{ib}(\xi, \, \boldsymbol{x}') * \boldsymbol{B}_{d,m}(\boldsymbol{x}'), \tag{9e}$$

$$\mathbb{Q}_{i,m}(\xi) = \mathbb{G}_{ii}(\xi, \, \xi') * \boldsymbol{A}_{i,m}(\xi'). \tag{9f}$$

3.2.3. Numerical approximation of the localization and interaction tensors

The prohibitive complexity related to the definition of analytical expressions for the localization and interaction tensors is circumvented by replacing them with their numerical approximations. Although other approaches can be used [20,21], here the FE method is adopted. To this end, the sub-domains and sub-interfaces are discretized using a fine FE mesh, termed the *support mesh*. For each sub-interface and sub-domain we define a set of sampling points $\boldsymbol{x}_h$ and ξ_h corresponding to FE nodes (Fig. 3).

The elastic localization solution (4) is obtained by solving the linear elastic problem defined by conditions (2)–(3) for the reference loads (i.e., for $\lambda = 1$) and by storing the strains and relative displacements at the sampling points

$$\boldsymbol{u}^*(\boldsymbol{x}) \sim \boldsymbol{u}^*(\boldsymbol{x}_h) \,, \quad \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \sim \boldsymbol{\varepsilon}^*(\boldsymbol{x}_h) \quad \text{and} \quad \boldsymbol{s}^*(\xi) \sim \boldsymbol{s}^*(\xi_h) \,, \tag{10}$$

where the symbol $\sim$ denotes here the replacement of a continuous variable with its sampling. The localized elastic solution for any load level can be obtained following (4) as

$$\check{\boldsymbol{u}}(\boldsymbol{x}) \sim \check{\boldsymbol{u}}(\boldsymbol{x}_h) = \lambda \boldsymbol{u}^*(\boldsymbol{x}_h) \,, \quad \check{\boldsymbol{\varepsilon}}(\boldsymbol{x}) \sim \check{\boldsymbol{\varepsilon}}(\boldsymbol{x}_h) = \lambda \boldsymbol{\varepsilon}^*(\boldsymbol{x}_h) \quad \text{and} \quad \check{\boldsymbol{s}}(\xi) \sim \check{\boldsymbol{s}}(\xi_h) = \lambda \boldsymbol{s}^*(\xi_h) \,. \tag{11}$$

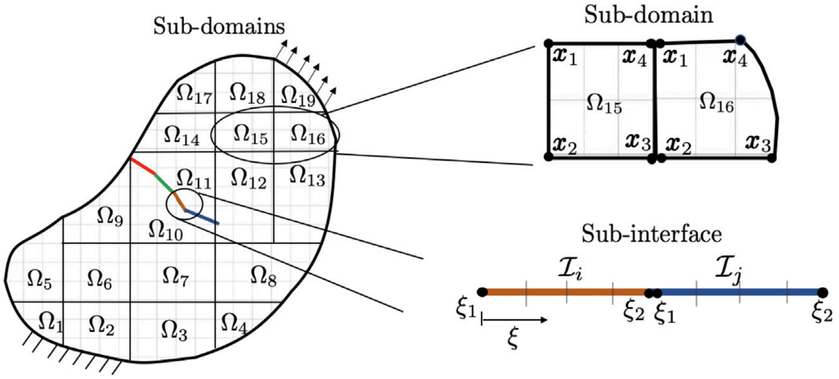


Fig. 3. Example of decomposition of the domain Ω in sub-domains and sub-interfaces based on a FE support mesh. The bulk domain is discretized in 19 sub-domains with 4 sampling points $\boldsymbol{x}_h$ and the interface $\mathcal{I}$ in 4 sub-interfaces with 2 sampling points ξ_h.

To obtain the numerical approximations $\boldsymbol{U}_{d,m}(\boldsymbol{x}_h)$, $\boldsymbol{T}_{i,m}(\boldsymbol{x}_h)$, $\boldsymbol{P}_{d,m}(\boldsymbol{x}_h)$, $\boldsymbol{S}_{i,m}(\boldsymbol{x}_h)$, $\boldsymbol{R}_{d,m}(\xi_h)$ and $\boldsymbol{Q}_{i,m}(\xi_h)$ of the interaction tensors (9), a set of linear FE analyses with homogeneous Dirichlet and Neumann boundary conditions is performed. Each computation involves the activation of a single component of each combination coefficient $\hat{\boldsymbol{\pi}}_{d,m}$ and $\hat{\boldsymbol{q}}_{i,m}$ with a unit excitation. The resulting distribution of inelastic deformations is imposed to the corresponding domain, and the obtained solution in terms of displacements, total strains and total relative displacements at the sampling points constitutes a single column of the approximation of (9).

When the interaction and localization tensors are approximated with the FE method, attention should be paid that all the quantities are defined at the nodes selected as sampling points. Here, since the displacements and their jump across the interfaces are naturally defined at the nodes, the matrices $\boldsymbol{U}_{d,m}(\boldsymbol{x}_h)$, $\boldsymbol{T}_{i,m}(\boldsymbol{x}_h)$, $\boldsymbol{R}_{d,m}(\xi_h)$ and $\boldsymbol{Q}_{i,m}(\xi_h)$ are readily available, while the strain components in $\boldsymbol{P}_{d,m}(\boldsymbol{x}_h)$ and $\boldsymbol{S}_{i,m}(\boldsymbol{x}_h)$ must be projected to the nodes. In both cases, the approximation of the tensors is performed in an offline phase before the beginning of the (online) incremental computation; it requires only a limited number of computationally inexpensive linear computations. The procedure to obtain the localization tensors is detailed in Appendix A, while the algorithm for the implementation of the offline phase is described in Appendix B.

3.3. Incremental problem and summary

The online part of the computation consists in the solution of the incremental version of the problem in Table 1, which reads as follows.

Problem (*Incremental Solution*)**.** Given $\lambda^{(k)}$ at load step k along with $\hat{\boldsymbol{\pi}}^{(k-1)}_{d,m}$ and $\hat{\boldsymbol{q}}^{(k-1)}_{i,m}$ at the previous load step $k-1$, determine $\hat{\boldsymbol{\pi}}^{(k)}_{d,m}$ and $\hat{\boldsymbol{q}}^{(k)}_{i,m}$, and consequently $\boldsymbol{u}^{(k)}(\boldsymbol{x}_h)$, $\boldsymbol{\varepsilon}^{(k)}(\boldsymbol{x}_h)$, $\boldsymbol{\sigma}^{(k)}(\boldsymbol{x}_h)$, $\boldsymbol{s}^{(k)}(\xi_h)$ and $\boldsymbol{t}^{(k)}(\xi_h)$ at load step k.

Assuming that the numerical approximations of the interaction and localization tensors are known from the offline phase, the solution is given as the superposition of (11) and (8), obtaining[1]

$$\boldsymbol{u}(\boldsymbol{x})^{(k)} \sim \boldsymbol{u}(\boldsymbol{x}_h)^{(k)} = \lambda^{(k)}\boldsymbol{u}^*(\boldsymbol{x}_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{U}_{d,m}(\boldsymbol{x}_h)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{T}_{i,m}(\boldsymbol{x}_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\,, \tag{12a}$$

$$\boldsymbol{\varepsilon}(\boldsymbol{x})^{(k)} \sim \boldsymbol{\varepsilon}(\boldsymbol{x}_h)^{(k)} = \lambda^{(k)}\boldsymbol{\varepsilon}^*(\boldsymbol{x}_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{P}_{d,m}(\boldsymbol{x}_h)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{S}_{i,m}(\boldsymbol{x}_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\,, \tag{12b}$$

$$\boldsymbol{\varepsilon}^{(k)}_e(\boldsymbol{x}) \sim \boldsymbol{\varepsilon}^{(k)}_e(\boldsymbol{x}_h) = \lambda^{(k)}\boldsymbol{\varepsilon}^*(\boldsymbol{x}_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \left(\boldsymbol{P}_{d,m}(\boldsymbol{x}_h) - \boldsymbol{B}_{d,m}(\boldsymbol{x}_h)\right)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{S}_{i,m}(\boldsymbol{x}_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\,, \tag{12c}$$

$$\boldsymbol{\sigma}^{(k)}(\boldsymbol{x}) \sim \boldsymbol{\sigma}^{(k)}(\boldsymbol{x}_h) = \mathbb{C}(\boldsymbol{x}_h) : \left[\lambda^{(k)}\boldsymbol{\varepsilon}^*(\boldsymbol{x}_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \left(\boldsymbol{P}_{d,m}(\boldsymbol{x}_h) - \boldsymbol{B}_{d,m}(\boldsymbol{x}_h)\right)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{S}_{i,m}(\boldsymbol{x}_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\right]\,, \tag{12d}$$

for the bulk, while for the interfaces

$$\boldsymbol{s}^{(k)}(\xi) \sim \boldsymbol{s}^{(k)}(\xi_h) = \lambda^{(k)}\boldsymbol{s}^*(\xi_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{R}_{d,m}(\xi_h)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{Q}_{i,m}(\xi_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\,, \tag{13a}$$

---

[1] It is worth recalling here that the inelastic deformation mode matrices $\boldsymbol{A}_{i,m}(\xi)$ and $\boldsymbol{B}_{d,m}(\boldsymbol{x})$ vanish outside the sub-domain and sub-interface they are associated with.

$$s_e^{(k)}(\xi) \sim s_e^{(k)}(\xi_h) = \lambda^{(k)} s^*(\xi_h) + \sum_{d=1}^{D} \sum_{m=1}^{M_\Omega} R_{d,m}(\xi_h) \hat{\pi}_{d,m}^{(k)} + \sum_{i=1}^{I} \sum_{m=1}^{M_\mathcal{I}} \left(Q_{i,m}(\xi_h) - A_{i,m}(\xi_h) \right) \hat{q}_{i,m}^{(k)} , \tag{13b}$$

$$t^{(k)}(\xi) \sim t^{(k)}(\xi_h) = \kappa(\xi_h) \cdot \left[\lambda^{(k)} s^*(\xi_h) + \sum_{d=1}^{D} \sum_{m=1}^{M_\Omega} R_{d,m}(\xi_h) \hat{\pi}_{d,m}^{(k)} + \sum_{i=1}^{I} \sum_{m=1}^{M_\mathcal{I}} \left(Q_{i,m}(\xi_h) - A_{i,m}(\xi_h) \right) \hat{q}_{i,m}^{(k)} \right] . \tag{13c}$$

In the previous expressions, using (7), the inelastic variable fields in (12)–(13) have been replaced by

$$\pi^{(k)}(x) \sim \pi^{(k)}(x_h) = \sum_{d=1}^{D} \sum_{m=1}^{M_\Omega} B_{d,m}(x_h) \hat{\pi}_{d,m}^{(k)} , \tag{14a}$$

$$q^{(k)}(\xi) \sim q^{(k)}(\xi_h) = \sum_{i=1}^{I} \sum_{m=1}^{M_\mathcal{I}} A_{i,m}(\xi_h) \hat{q}_{i,m}^{(k)} . \tag{14b}$$

3.4. Available solution schemes

From (12)–(13) it is clear that the unknowns of the problem consist in the inelastic combination coefficients $\hat{\pi}_{d,m}^{(k)}$ and $\hat{q}_{i,m}^{(k)}$. A first strategy to define them is proposed in [11], where a fixed point iterative scheme is adopted to determine the combination coefficients related to the assumed piecewise uniform deformation modes. The approach is extended to non-uniform ansatz functions in [12,13] by introducing a sampling of the inelastic relative displacements and strains in a predefined subset of sampling points within each sub-domain and sub-interface, using the evolution equations in Table 1 along with (12b) and (13a). The algorithmic iterative update of the combination coefficients is then computed through a least square problem, where the objective function to be minimized is the discrepancy between the sampled values and their representations using (7). To this end, the number of sampling points in the subset must be equal to or larger than M_Ω and $M_\mathcal{I}$ for the sub-domains and sub-interfaces, respectively. In this case, the combination coefficients have merely the meaning of best-fitting parameters and their evolution does not comply with the evolution equations of Table 1. Also, we remark that the combination coefficients $\hat{\pi}_{d,m}$ and $\hat{q}_{i,m}$ are defined individually for each subdivision and, hence, the parameterization (7) is not continuous at the border between neighboring sub-domains and sub-interfaces.

A drawback related to the least square fitting procedure is that it prevents the possibility to adopt a computationally more efficient Newton–Raphson scheme based on the consistent tangent matrix; in turn, such scheme would enable the employment of an arc-length procedure to follow possible unstable branches in the equilibrium path, e.g. in case of snap-backs. Motivated by such observations, we devise here a strategy with allows for the definition of a consistent tangent matrix. We term this strategy *Enhanced Transformation Field Analysis (E-TFA)*.

4. Enhanced Transformation Field Analysis (E-TFA)

This section describes the derivation of the consistent tangent matrix for E-TFA, which enables the adoption of Newton–Raphson and arc-length solution schemes.

4.1. Selection of the sampling points and deformation modes

We start by selecting a set of $M_\Omega = (p+1)\times(p+1)$ and $M_\mathcal{I} = (p+1)$ sampling points, x_h and ξ_h respectively, for every sub-domain and sub-interface, where p is the order of polynomials used for approximating the inelastic fields.[2] The inelastic modes are selected as the Lagrange polynomials with nodes coinciding with the sampling points, namely, the nth mode takes a unit value at the nth sampling point and vanishes at the others. The matrices $A_{i,m}(\xi)$ and $B_{d,m}(x)$ can thus be defined in Voigt notation as

$$B_{d,m}(x) = \text{diag}\left\{ b_m(x),\, b_m(x),\, b_m(x) \right\} \quad \text{and} \quad A_{i,m}(\xi) = \text{diag}\left\{ a_m(\xi),\, a_m(\xi) \right\} \tag{15}$$

where $b_m(x)$ and $a_m(\xi)$ are the mth modes related to each degree of freedom for the dth sub-domain and ith sub-interface, respectively. Hence, the combination coefficients for the mth mode become

$$\hat{\pi}_{d,m} = \pi(x_m) \quad \text{and} \quad \hat{q}_{i,m} = q(\xi_m) . \tag{16}$$

Unlike in the approach proposed in [12,13], this choice grants physical meaning to the combination coefficients and establishes a direct link between them and the evolution equations in Table 1,[3] while removing the need for a least square minimization problem enabling the definition of the consistent tangent matrix.

---

[2] For simplicity, we assume here a unique value of p for all sub-domains and sub-interfaces; however, in principle the polynomial degree can be selected individually for each sub-domain and sub-interface depending on the expected complexity of the inelastic deformations.

[3] In a different but fully equivalent form, (15) can be written adopting ansatz functions defined globally on the sub-domain instead of associating them to the sampling points in a FE-like fashion. In this case, the approximations (7) must be defined as

$$\pi(x) \simeq \sum_{d=1}^{D} \tilde{B}_d(x) \hat{\pi}_d^* \quad \text{and} \quad q(\xi) \simeq \sum_{i=1}^{I} \tilde{A}_i(\xi) \hat{q}_i^* ,$$

where the entries of the inelastic mode matrices are the monomials composing the Lagrange polynomials and the combination coefficients for a 2D problem with dimensions $(1 \times 2M_\mathcal{I})$ and $(1 \times 3M_\Omega)$ for the sub-interfaces and the sub-domains, respectively.

4.2. Consistent tangent matrix

To define the consistent tangent matrix we write the residuals at load step k for the sub-domains and the sub-interfaces using (12b) and (13a) as

$$\boldsymbol{r}_{\Omega}(\boldsymbol{x}_h)^{(k)} = \boldsymbol{\varepsilon}(\boldsymbol{x}_h)^{(k)} - \left[\lambda^{(k)}\boldsymbol{\varepsilon}^{*}(\boldsymbol{x}_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_{\Omega}} \boldsymbol{P}_{d,m}(\boldsymbol{x}_h)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_{\mathcal{I}}} \boldsymbol{S}_{i,m}(\boldsymbol{x}_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\right], \tag{17a}$$

$$\boldsymbol{r}_{\mathcal{I}}(\xi_h)^{(k)} = \boldsymbol{s}^{(k)}(\xi_h) - \left[\lambda^{(k)}\boldsymbol{s}^{*}(\xi_h) + \sum_{d=1}^{D}\sum_{m=1}^{M_{\Omega}} \boldsymbol{R}_{d,m}(\xi_h)\hat{\boldsymbol{\pi}}^{(k)}_{d,m} + \sum_{i=1}^{I}\sum_{m=1}^{M_{\mathcal{I}}} \boldsymbol{Q}_{i,m}(\xi_h)\hat{\boldsymbol{q}}^{(k)}_{i,m}\right], \tag{17b}$$

respectively. They can be casted in the following matrix form

$$\boldsymbol{r}^{(k)} = \begin{vmatrix}\boldsymbol{r}^{(k)}_{\Omega}\\ \boldsymbol{r}^{(k)}_{\mathcal{I}}\end{vmatrix} = \begin{vmatrix}\boldsymbol{\varepsilon}^{(k)}\\ \boldsymbol{s}^{(k)}\end{vmatrix} - \left[\lambda^{(k)}\begin{vmatrix}\boldsymbol{\varepsilon}^{*}\\ \boldsymbol{s}^{*}\end{vmatrix} + \begin{bmatrix}\boldsymbol{P} & \boldsymbol{S}\\ \boldsymbol{R} & \boldsymbol{Q}\end{bmatrix}\begin{vmatrix}\hat{\boldsymbol{\pi}}^{(k)}\\ \hat{\boldsymbol{q}}^{(k)}\end{vmatrix}\right], \tag{18}$$

where the dependence on the spatial coordinate is dropped for simplicity.[4] $\boldsymbol{P}$, $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{Q}$ are global matrices collecting all the interaction tensors and $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{q}}$ are global vectors storing all the combination coefficients. The detailed definition of each component of (18) is shown in Appendix A.

To formulate a Newton–Raphson iterative scheme, the residuals at the nth iteration are expanded to first order as

$$\boldsymbol{r}^{(k,n)} = \boldsymbol{r}^{(k,n-1)} + \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}^{(k,n-1)}} \cdot \Delta\boldsymbol{\varepsilon}^{(k,n)} + \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}^{(k,n-1)}} \cdot \Delta\boldsymbol{s}^{(k,n)}, \tag{19}$$

where $\Delta\boldsymbol{\varepsilon}^{(k,n)}$ and $\Delta\boldsymbol{s}^{(k,n)}$ are the iterative corrections to the total strains and interface relative displacements, respectively. Using the matrix form (18), we have

$$\boldsymbol{r}^{(k,n)} = \boldsymbol{r}^{(k,n-1)} + \underbrace{\begin{bmatrix}\boldsymbol{I}_{\varepsilon} - \boldsymbol{P}\left.\dfrac{\partial\hat{\boldsymbol{\pi}}}{\partial\boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}^{(k,n-1)}} & -\boldsymbol{S}\left.\dfrac{\partial\hat{\boldsymbol{q}}}{\partial\boldsymbol{s}}\right|_{\boldsymbol{s}^{(k,n-1)}}\\ -\boldsymbol{R}\left.\dfrac{\partial\hat{\boldsymbol{\pi}}}{\partial\boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}^{(k,n-1)}} & \boldsymbol{I}_{s} - \boldsymbol{Q}\left.\dfrac{\partial\hat{\boldsymbol{q}}}{\partial\boldsymbol{s}}\right|_{\boldsymbol{s}^{(k,n-1)}}\end{bmatrix}}_{\boldsymbol{K}^{(k,n)}}\begin{vmatrix}\Delta\boldsymbol{\varepsilon}^{(k,n)}\\ \Delta\boldsymbol{s}^{(k,n)}\end{vmatrix}, \tag{20}$$

where $\boldsymbol{I}_{\varepsilon}$ and $\boldsymbol{I}_{s}$ are the identity matrices with dimension equal to the total number of degrees of freedom for the strains and interface relative displacements, respectively. The matrix indicated with $\boldsymbol{K}^{(k,n)}$ is the consistent tangent matrix, which is full and unsymmetric; it depends on the derivatives of the combination coefficients with respect to strains and interface relative displacements. Thanks to (16), the latter can be calculated directly from the adopted evolution equations and the complete expressions for the case of Table 1 are reported in Appendix C.

At each iteration the corrections to the total strains and interface relative displacements can be obtained from (20) as

$$\boldsymbol{r}^{(k,n)} \overset{!}{=} 0 \quad\Rightarrow\quad \begin{vmatrix}\Delta\boldsymbol{\varepsilon}^{(k,n)}\\ \Delta\boldsymbol{s}^{(k,n)}\end{vmatrix} = -\left(\boldsymbol{K}^{(k,n)}\right)^{-1}\boldsymbol{r}^{(k,n-1)}, \tag{21}$$

thus

$$\boldsymbol{\varepsilon}^{(k,n)} = \boldsymbol{\varepsilon}^{(k,n-1)} + \Delta\boldsymbol{\varepsilon}^{(k,n)}, \tag{22a}$$

$$\boldsymbol{s}^{(k,n)} = \boldsymbol{s}^{(k,n-1)} + \Delta\boldsymbol{s}^{(k,n)}, \tag{22b}$$

$$\hat{\boldsymbol{\pi}}^{(k,n)} = \hat{\boldsymbol{\pi}}^{(k,n-1)} + \left.\frac{\partial\hat{\boldsymbol{\pi}}}{\partial\boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}^{(k,n-1)}} \cdot \Delta\boldsymbol{\varepsilon}^{(k,n)} = \hat{\boldsymbol{\pi}}^{(k,n-1)} + \Delta\hat{\boldsymbol{\pi}}^{(k,n)}, \tag{22c}$$

$$\hat{\boldsymbol{q}}^{(k,n)} = \hat{\boldsymbol{q}}^{(k,n-1)} + \left.\frac{\partial\hat{\boldsymbol{q}}}{\partial\boldsymbol{s}}\right|_{\boldsymbol{s}^{(k,n-1)}} \cdot \Delta\boldsymbol{s}^{(k,n)} = \hat{\boldsymbol{q}}^{(k,n-1)} + \Delta\hat{\boldsymbol{q}}^{(k,n)}, \tag{22d}$$

where $\Delta\hat{\boldsymbol{\pi}}^{(k,n)}$ and $\Delta\hat{\boldsymbol{q}}^{(k,n)}$ are the iterative corrections to non-linear strains and relative displacements, respectively.

Note that the online computation of the consistent tangent matrix does not involve any numerical integration or assembly operation, since it consists in the addition of factors given by the product between matrices stored during the offline phase and

---

[4] Note that in the residual vector $\boldsymbol{r}^{(k)}$ in (18), the two subvectors $\boldsymbol{r}^{(k)}_{\Omega}$ and $\boldsymbol{r}^{(k)}_{\mathcal{I}}$ include different quantities, namely strains and relative displacements, respectively. To solve this inconsistency here and in the related quantities presented in the following, the residual vector related to the relative displacements $\boldsymbol{r}^{(k)}_{\mathcal{I}}$ can be normalized with a characteristic interface length, such as an interface thickness.

the derivative of the combination coefficients with respect to strains and interface relative displacements. This feature, along with the reduction of the number of unknowns obtained by grouping different finite elements per sub-domain or sub-interface, makes the approach more efficient compared to the FE method. On the other hand, the inversion of the tangent stiffness matrix is computationally more expensive due to its dense nature; hence, for refined subdivisions where each sub-domain and sub-interface contain only one or few finite elements, E-TFA is no longer competitive.

4.3. Arc-length approach

The definition of the consistent tangent matrix (20) enables an arc-length solution method to track possible snap-back branches. Although more advanced procedures are available [27], we base our approach on two well-established arc-length strategies: the spherical arc-length proposed by Crisfield [28] and its linearization proposed by Ramm [29]. To this end, we treat the load multiplier as a variable that needs to be updated at every iteration, namely $\lambda = \lambda^{(k,n)}$. Hence, the linearized residual (19) becomes

$$\begin{aligned} \boldsymbol{r}^{(k,n)} &= \boldsymbol{r}^{(k,n-1)} + \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}^{(k,n-1)}} \cdot \Delta\boldsymbol{\varepsilon}^{(k,n)} + \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}^{(k,n-1)}} \cdot \Delta\boldsymbol{s}^{(k,n)} + \left.\frac{\partial \boldsymbol{r}}{\partial \lambda}\right|_{\lambda^{(k,n-1)}} \cdot \Delta\lambda^{(k,n)} \\ &= \boldsymbol{r}^{(k,n-1)} + \boldsymbol{K}^{(k,n)} \left|\begin{matrix} \Delta\boldsymbol{\varepsilon}^{(k,n)} \\ \Delta\boldsymbol{s}^{(k,n)} \end{matrix}\right| - \Delta\lambda^{(k,n)} \left|\begin{matrix} \boldsymbol{\varepsilon}^{*} \\ \boldsymbol{s}^{*} \end{matrix}\right| , \end{aligned} \tag{23}$$

where $\Delta\lambda^{(k,n)}$ is the iterative correction to the load multiplier. Solving for the iterative corrections gives

$$\underbrace{\left|\begin{matrix} \Delta\boldsymbol{\varepsilon}^{(k,n)} \\ \Delta\boldsymbol{s}^{(k,n)} \end{matrix}\right|}_{\Delta\boldsymbol{y}^{(k,n)}} = \underbrace{-\left(\boldsymbol{K}^{(k,n)}\right)^{-1} \boldsymbol{r}^{(k,n-1)}}_{\Delta\boldsymbol{y}_{r}^{(k,n)}} + \Delta\lambda^{(k,n)} \underbrace{\left(\boldsymbol{K}^{(k,n)}\right)^{-1} \left|\begin{matrix} \boldsymbol{\varepsilon}^{*} \\ \boldsymbol{s}^{*} \end{matrix}\right|}_{\Delta\boldsymbol{y}_{\lambda}^{(k,n)}} , \tag{24}$$

Here $\Delta\boldsymbol{y}^{(k,n)}$ collects the total iterative corrections to the strain and interface relative displacements and $\Delta\boldsymbol{y}_{r}^{(k,n)}$ and $\Delta\boldsymbol{y}_{\lambda}^{(k,n)}$ are quantities associated with the residual and the load multiplier, respectively.

Crisfield's arc-length. Considering λ as a further variable calls for an additional equation termed *arc-length constraint* which, in the spirit of [28], is defined as

$$\left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_{c}^{(k,l)} + \Delta\boldsymbol{y}_{c}^{(k,n)}\right) \cdot \left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_{c}^{(k,l)} + \Delta\boldsymbol{y}_{c}^{(k,n)}\right) - \ell^{2} = 0 , \tag{25}$$

where the subscript c indicates the restriction of the corresponding vector to a proper subset of *control degrees of freedom* whose selection is clarified in the following. Also, ℓ is the *arc-length radius* parameter and, at the first iteration, we assume $\Delta\boldsymbol{y}_{c}^{(k,0)} = \boldsymbol{0}$.

Combining (24) and (25), we obtain the following quadratic equation in the unknown $\Delta\lambda^{(k,n)}$

$$C_2 \left(\Delta\lambda^{(k,n)}\right)^2 + C_1 \Delta\lambda^{(k,n)} + C_0 = 0 , \tag{26}$$

with the scalar coefficients C_2, C_1 and C_0 given by

$$\begin{aligned} C_2 &= \Delta\boldsymbol{y}_{\lambda,c}^{(k,n)} \cdot \Delta\boldsymbol{y}_{\lambda,c}^{(k,n)} , \\ C_1 &= 2\left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_{c}^{(k,l)} + \Delta\boldsymbol{y}_{r,c}^{(k,n)}\right) \cdot \Delta\boldsymbol{y}_{\lambda,c}^{(k,n)} , \\ C_0 &= \left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_{c}^{(k,l)} + \Delta\boldsymbol{y}_{r,c}^{(k,n)}\right) \cdot \left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_{c}^{(k,l)} + \Delta\boldsymbol{y}_{r,c}^{(k,n)}\right) - \ell^{2} . \end{aligned} \tag{27}$$

The arc-length radius is defined at the beginning of the first load step by setting $\lambda^{(1,1)} = \Delta\lambda^{(1,1)} = \lambda^{*}$, where λ^{*} is a free parameter. For the first iteration the residual is identically zero and, using (24), we obtain $\Delta\boldsymbol{y}_{r}^{(1,1)} = \boldsymbol{0}$ and $\Delta\boldsymbol{y}^{(1,1)} = \lambda^{*}\Delta\boldsymbol{y}_{\lambda}^{(1,1)}$. Hence, the arc-length radius can be computed inverting (26) under the assumption that $\Delta\boldsymbol{y}_{c}^{(1,1)} \equiv \Delta\boldsymbol{y}^{(1,1)}$

$$\ell = \lambda^{*} \sqrt{\Delta\boldsymbol{y}^{(1,1)} \cdot \Delta\boldsymbol{y}^{(1,1)}} . \tag{28}$$

This value is then kept constant for the whole computation.

For each load step $k \geq 2$, the first correction to the load multiplier $\Delta\lambda^{(k,1)}$ is computed considering that $\Delta\boldsymbol{y}_{r}^{(k,1)} = \boldsymbol{0}$, hence

$$\Delta\lambda^{(k,1)} = \left| \frac{\ell}{\sqrt{\Delta\boldsymbol{y}_{\lambda}^{(k,1)} \cdot \Delta\boldsymbol{y}_{\lambda}^{(k,1)}}} \right| . \tag{29}$$

The existence of two roots for (26) calls for a method to distinguish which one is correct for the prescribed monotonic loading. We adopt here the simple but effective criterion proposed in [28], where the root is chosen as

$$\Delta\lambda^{(k,n)} = \arg\max_{\Delta\lambda} \left\{ \left(\sum_{l=0}^{n-1} \Delta\boldsymbol{y}_c^{(k,l)} \right) \cdot \left(\sum_{l=0}^{n} \Delta\boldsymbol{y}_c^{(k,l)} \right), \quad \text{subject to (24)–(27)} \right\} . \tag{30}$$

Control subset. The discriminant of the quadratic equation (26) is not necessarily positive, however, a proper definition of the control degrees of freedom greatly limits the occurrence of imaginary roots [30,31]. In particular, within monotonic loading, a stable process can be achieved by selecting as control subset all the degrees of freedom whose evolution at a given load induces an increase of the energy dissipation. In other words, the degrees of freedom that undergo elastic loading/unloading/reloading must be excluded from the control pool.

The development of a procedure to fully comply with this strategy is out of the scope of the present paper; here we apply an approximate but effective procedure. Considering that the occurrence of snap-back branches is due to the presence of cohesive interfaces with a marked stress-softening behavior, we include in the control pool only the interface relative displacements. Moreover, a degree of freedom related to an interface relative displacement is considered to be contributing to the increase of dissipation and therefore included in the control subset for the load step k if it complies with the following criterion

$$s_T^{(k,n)}(\xi_h) \in \boldsymbol{y}_c^{(k,n)} \ \text{ if } \ \left| s_T^{(k-1)}(\xi_h) \right| \geq \max_{t \leq k-2} \left\{ \left| s_T^{t}(\xi_h) \right| \right\} , \tag{31}$$

in tangential direction and/or

$$s_N^{(k,n)}(\xi_h) \in \boldsymbol{y}_c^{(k,n)} \ \text{ if } \ s_N^{(k-1)}(\xi_h) \geq 0 \ \& \ s_N^{(k-1)}(\xi_h) \geq \max_{t \leq k-2} \left\{ s_N^{t}(\xi_h) \right\} , \tag{32}$$

in normal direction. Although this approach does not strictly ensure that (26) has real roots, it proved reliable for the examples in this paper.

Ramm's arc-length. Especially in presence of many interfaces it may happen that, even applying the criteria (31)–(32), the roots of (26) are imaginary. Although different approaches have been proposed in the literature to amend this well-known issue [28,32,33], no universal solution is available. Here, we adopt the linearized arc-length strategy proposed by Ramm [29].

In Ramm's approach the definition of the arc-length radius and of the first iterative correction of the load multiplier is the same as in Crisfield's arc-length. However, for the subsequent load multiplier updates the following constraint equation is used instead of (26)

$$\Delta\boldsymbol{y}_c^{(k,1)} \cdot \Delta\boldsymbol{y}_c^{(k,n)} = 0 \quad \text{for } n \geq 2 , \tag{33}$$

namely, the iterative correction to the control subset must be orthogonal to the first one. Hence, the load multiplier correction becomes

$$\Delta\lambda^{(k,n)} = -\frac{\Delta\boldsymbol{y}_c^{(k,1)} \cdot \Delta\boldsymbol{y}_{r,c}^{(k,n)}}{\Delta\boldsymbol{y}_c^{(k,1)} \cdot \Delta\boldsymbol{y}_{\lambda,c}^{(k,n)}} \quad \text{for } n \geq 2 . \tag{34}$$

The drawback of this approach is that the magnitude of the iterative corrections are limited to the arc-length radius only for the first iteration, while for the subsequent ones there is no limit. This may prevent convergence in the vicinity of limit points [34], however this never happened for the tests illustrated here.

4.4. Displacement field and comparison with FE results

Once the combination coefficient are known, they can be introduced in (12) and (13) to obtain the main relevant quantities. Thanks to the assumptions made in Section 4.1, the inelastic modes can be used similarly to the shape functions in the FE method to approximate the spatial distribution of the fields in (12)–(13). An alternative strategy may be to project the approximate inelastic deformations $\boldsymbol{\pi}^{(k)}(\boldsymbol{x})$ and $\boldsymbol{q}^{(k)}(\xi)$ to the FE space using (7), delivering the (approximate) inelastic variables at the Gauss points. The main mechanical fields are then obtained performing a linear FE computation, where the assembled stiffness matrix represents an approximation of the secant stiffness of the system at a given load level $\lambda^{(k)}$. This allows for a finer reconstruction of the main fields.

Comparing the FE and E-TFA approaches to solve the problem described in Section 2, we remark that in the former case the tangent stiffness matrix results sparse and includes unsymmetric entries only when the interfacial frictional sliding is active (C.3b). Conversely, the E-TFA case involves a full tangent stiffness matrix that is unsymmetric by construction due to the unsymmetric nature of the interaction tensors $\boldsymbol{P}$, $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{Q}$ along with the unsymmetric off-diagonal contributions in (20).

Note that, if the non-linearities are concentrated at the interfaces (i.e., the bulk is linearly elastic), only the latter need to be discretized into sub-interfaces for the computation of the interaction tensors since the combination coefficients for the bulk inelastic strains are zero, namely $\hat{\boldsymbol{\pi}}_{d,m} = \boldsymbol{0}$ and only $\hat{\boldsymbol{q}}_{i,m} \neq \boldsymbol{0}$. Moreover, if each sub-interface spans a single finite element of the support mesh, the E-TFA solution coincides with the FE solution, provided that the inelastic modes and shape functions are the same. In this case, E-TFA is expected to be more efficient than FE because the Newton–Raphson updates involve only the inelastic coefficients of the interfaces, leading to a considerable reduction of the tangent matrix size. Concerning the bulk material, the solution is conveniently obtained using (12)–(13) in a post-processing phase involving only matrix additions and multiplications.

When also the bulk is inelastic, the difference between E-TFA and FE solutions is that the FE solution in the bulk is given in terms of displacements and the inelastic variables are defined at the Gauss points, while in E-TFA the primary variables are the inelastic deformations computed at the sampling points (which coincide with the nodes of the support mesh).

5. Numerical examples

In this section we demonstrate the capability of the proposed approach to solve problems of the type presented in Section 2 in a computationally efficient yet accurate way. In particular, we compare the E-TFA results with reference solutions obtained with the FE method. Both FE and E-TFA analyses are performed under plane strain conditions and reported per unit thickness (if not specified otherwise). The codes are based on the C++ library *deal.II* [35] and the computations are performed on a single core of the Euler High Performance Computing cluster of ETH Zürich.

5.1. Numerical aspects

The domain is first divided into sub-domains and sub-interfaces and then further discretized into a support mesh composed of bilinear quadrilateral and linear interface finite elements. We remark that the library *deal.II* [35] does not allow for a straightforward implementation of zero-thickness interfaces that, therefore, have a negligible thickness of 10^{-3} mm and are implemented following [36]. The inelastic modes for the sub-interfaces and sub-domains are assumed to be Lagrange polynomials of degree $p = 1$, namely linear and bilinear ansatz functions are adopted. This leads to selecting four and two sampling points respectively for each sub-domain (i.e., $M_{\Omega} = 4$) and sub-interface (i.e., $M_{\mathcal{I}} = 2$), which are here taken as the corner nodes of the underlying support mesh (Fig. 3). To facilitate the node selection, the sub-domains are defined with a quadrilateral shape.

For the iterative solution of the E-TFA problem during the online phase (Appendix B) we use the biconjugate gradient stabilized (*BICGSTAB*) method [37] to invert the full and unsymmetric consistent tangent matrix. For the reference FE analyses, the interface elements are implemented following [36], while for the solution of the system a standard iterative Newton–Raphson scheme based on the direct sparse solver unsymmetric-pattern multifrontal method *UMFPACK* [38] is adopted. Concerning the arc-length approach illustrated in Section 4.3, its FE version is implemented following [29,30]. The attained speed-up is evaluated as the ratio between the wall time taken by the FE and E-TFA analyses. For the latter, the share of wall time needed for the online and offline phases is indicated as well.

For both E-TFA and FE computations, the solution is considered converged when the L2-norm of the residual vector reaches a predefined tolerance, namely when $\left\| \boldsymbol{r}^{(k,n)} \right\| \leq tol$. The E-TFA residuals are defined in (18), while for FE they are given by the customary vector of residual forces. Owing to the different nature of the FE and E-TFA residual vectors, the definition of a tolerance selection criterion to ensure a fair comparison of the results is not trivial. Here, we adopt empirical choices by studying convergence of the solution with varying tolerance values for both approaches. Clearly, the choice of the tolerance impacts the number of iterations in the non-linear solution procedure, which in general influences the speed-up achieved by E-TFA over FE.

Whenever we do not use the arc-length approach, we adopt a unique absolute value of the tolerance $tol = 1\text{e}{-5}$, which guarantees converged solutions from both FE and E-TFA. For the cases with arc-length, it is not possible to establish a single tolerance for all the computations because the magnitude of the corrections to the solution is bounded by the arc-length radius ℓ. This is determined at the beginning of each analysis using (28), hence, it is a function of the first correction of the solution vector that can assume different values for the E-TFA and FE analyses. Therefore, in such computations we prescribe the tolerance as a function of the arc-length radius as $tol = \ell/100$.

5.2. Four inclusions in a linear elastic matrix

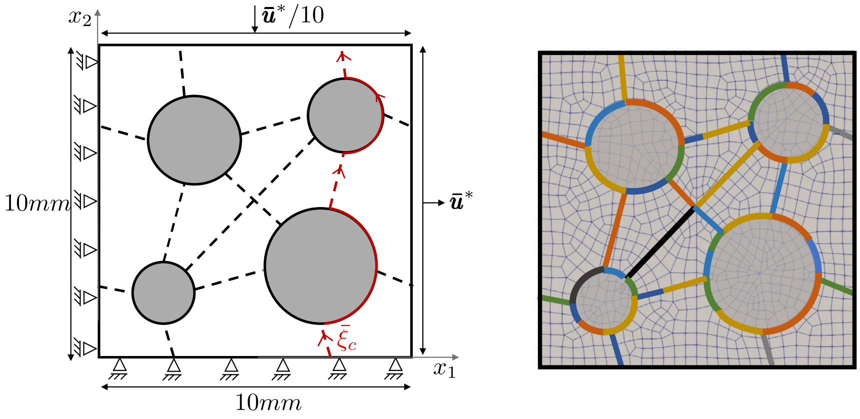
In this example, we study a square domain containing four irregularly distributed inclusions under biaxial loading conditions. Fig. 4(a) shows the geometry and boundary conditions, along with the arrangement of the aggregate-matrix (solid lines) and matrix–matrix (dashed lines) interfaces. The aim of this example is to show the capabilities of the proposed E-TFA arc-length approach (Section 4.3), hence, the matrix is assumed linearly elastic and the material parameters (summarized in Table 2) are tuned so as to induce multiple snap-backs. The reference loading parameter is $\bar{u}^{*} = 10^{-2}$ mm.

The details of the E-TFA and FE discretizations are summarized in Table 3. Since here the bulk is linearly elastic, in E-TFA we do not need to introduce sub-domains but only sub-interfaces (see Section 4.4). We consider a coarse, a medium and a 1:1 subdivision, where the first is purposefully very rough, the second is obtained by uniformly refining the first one and the third contains as many sub-interfaces as in the FE mesh. The coarse subdivision is illustrated in Fig. 4(b) for reference. With this comparison we aim at investigating convergence of E-TFA results towards FE results by refining the interface subdivision. Fig. 4(b) also shows the background mesh which coincides with the FE mesh and is kept the same in all simulations.

Fig. 5(a) illustrates the reaction force vs. displacement curves in horizontal direction. It is clear that the proposed arc-length approach is able to follow multiple snap-backs while preserving an excellent speed-up (Table 3). We can also see that the E-TFA results converge towards the FE results as the subdivision is refined. This is further confirmed by Fig. 6, where the inelastic contributions to the interface relative displacement for the load steps A, B and C in Fig. 5(a) are compared along the set of interfaces

Table 2
Material parameters adopted for the biaxial test on a specimen with four inclusions.

Parameter	Bulk		Interfaces	
	Inclusions	Matrix	Matrix-inclusion	Matrix–matrix
E (GPa)	300	70	–	–
ν (–)	0.2	0.3	–	–
$\bar{t}_T$ (MPa)	–	–	300	300
$\bar{t}_N$ (MPa)	–	–	300	300
$\bar{G}_{c,T}$ (N/mm)	–	–	2.0	8.0
$\bar{G}_{c,N}$ (N/mm)	–	–	2.0	8.0
k_T (GPa/mm)	–	–	1000	1000
k_N (GPa/mm)	–	–	1000	1000
μ (–)	–	–	0.5	0.5



(a) (b)

Fig. 4. (a) Geometry and boundary conditions adopted for the biaxial test on a specimen with four inclusions and (b) background mesh and sub-interface subdivision for the coarse E-TFA discretization (different colors identify different sub-interfaces). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Details of the numerical computations for the biaxial test on a specimen with four inclusions.

	FE	E-TFA			
		Coarse	Medium	1:1	Selectively refined
Interfaces	176 elements	43 sub-interfaces	75 sub-interfaces	176 sub-interfaces	102 sub-interfaces
CPU time	100% online	~98% online ~2% offline	~97.5% online ~2.5% offline	~97.5% online ~2.5% offline	~96.8% online ~3.2% offline
Speed-up	1 (Reference)	~24×	~21×	~11×	~13×

highlighted in red in Fig. 4(a). The accuracy of the inelastic field steadily increases as the subdivision is refined and, for the 1:1 case, a perfect match is obtained. Note that this still comes with an improvement of the computational time of about 11 times. The computational gain in this case is due to the reduction of the number of variables as well as to a faster convergence of the iterative solution in the non-linear load steps. The latter behavior is demonstrated in Fig. 5(b), where the number of iterations required per load step for E-TFA (regardless of the subdivision) is almost one order of magnitude smaller than for FE. This is due to the fact that the variables used to drive the arc-length strategy are the interface relative displacements, which are the primary – and in this case the only – variables for E-TFA. This makes the implementation of the arc-length strategy within the E-TFA approach straightforward, fast and very robust. Conversely, for the FE method, the main variables are the nodal displacements of the whole domain, while the interface relative displacements need to be post-processed from a subset of the degrees of freedom.

The similarities with the FE results in Figs. 5(a) and 6 suggest that, although with a certain approximation, the coarse and medium subdivisions are able to capture the main interface degradation mechanisms under loading. In particular, comparing the displacement field in horizontal direction at load steps A and C (Fig. 7a–c) we notice that interfacial decohesion always starts at the first snap-back (step A) along the right sub-vertical interfaces and around the upper right inclusion. Nevertheless, the lower accuracy in reproducing the inelastic fields leads to a slightly different final failure mechanism, which is demonstrated by the difference between FE and E-TFA results in Fig. 8. This difference can be explained considering that the state depicted in Fig. 7a–c for load

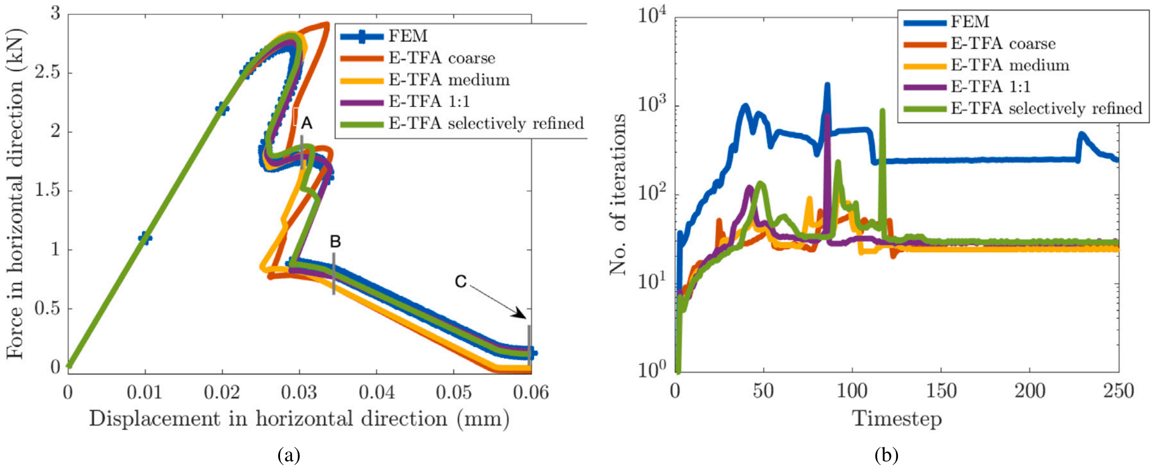


Fig. 5. (a) Load–displacement curves and (b) iterations required per time step for the biaxial test on a specimen with four inclusions.

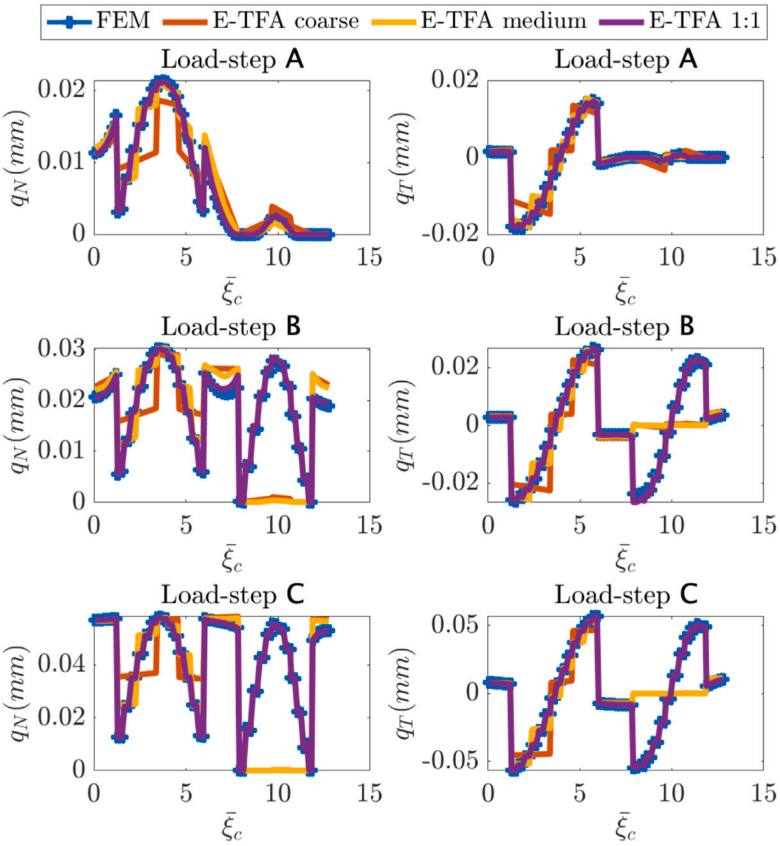


Fig. 6. Inelastic relative displacements in normal and tangential direction along the collective curvilinear coordinate $\bar{\xi}_c$ defined in Fig. 4(a).

step A, where all the interfaces around the upper right inclusion are in the softening stage, is unstable. Hence, small perturbations may lead the debonding process of one side to prevail on the other.

Based on this observation, we define a new subdivision which is selectively refined along the right sub-vertical interfaces and around the upper right inclusion, leading to a total of 102 sub-interfaces (Table 3). The results obtained with this subdivision are reported in Figs. 7 and 5(a)e, and show an agreement similar to the 1:1 subdivision while reaching a speed-up of about 13 times, i.e. about 16% faster than the 1:1 case (Table 3). Clearly, in order to know where to refine the interface subdivision, we need to

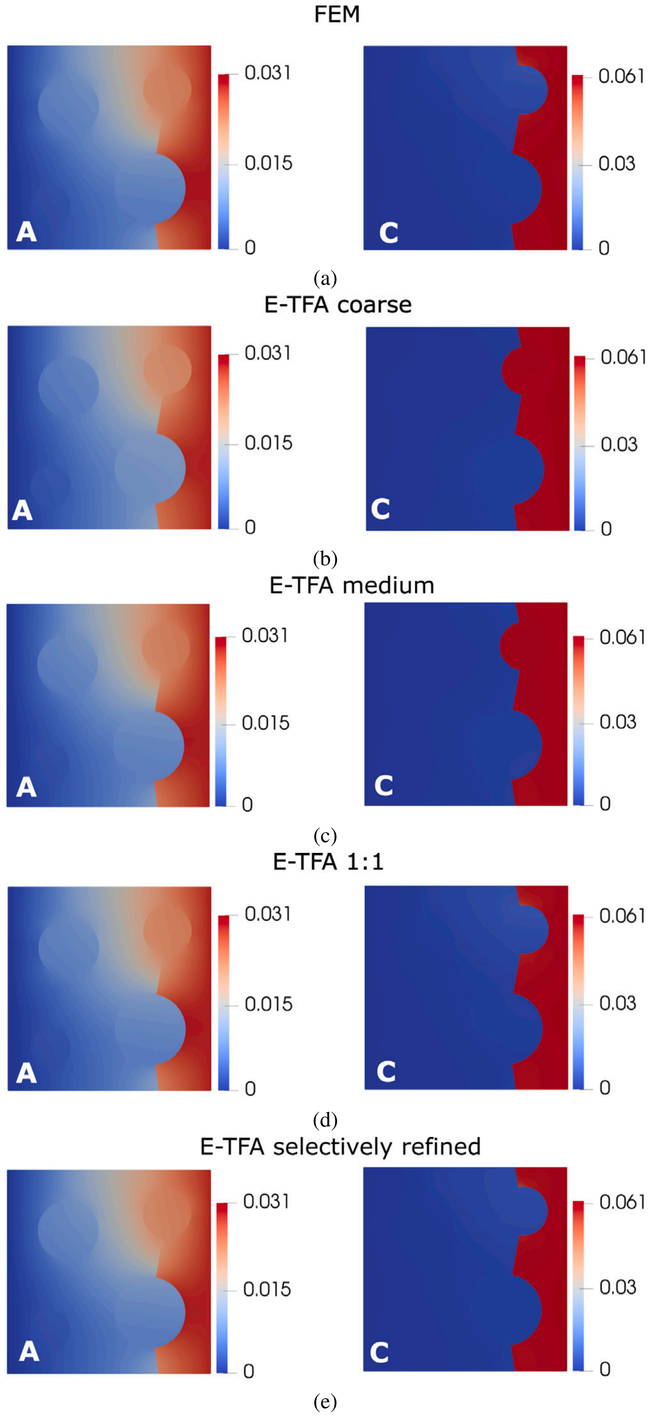


Fig. 7. Horizontal displacement field for the biaxial test on a specimen with four inclusions at load steps A and C (highlighted in Fig. 5(a)) from (a) FE and E-TFA with (b) coarse, (c) medium, (d) 1:1 and (e) selectively refined interfaces.

first run a coarse or medium analysis. But even including the computational cost for the coarse (medium) analysis, we still obtain a speed-up of about 8.3 (8) times with respect to the FE analysis. Moreover, the selectively refined subdivision captures also the correct failure mechanism (Fig. 7d,e), which is not obtained even with the medium subdivision, although a similar number of sub-interfaces is used. Despite involving two separate analyses, this strategy is convenient thanks to the negligible computational time required by the coarser E-TFA analysis that allows to determine which interfaces play a leading role in the final behavior and,

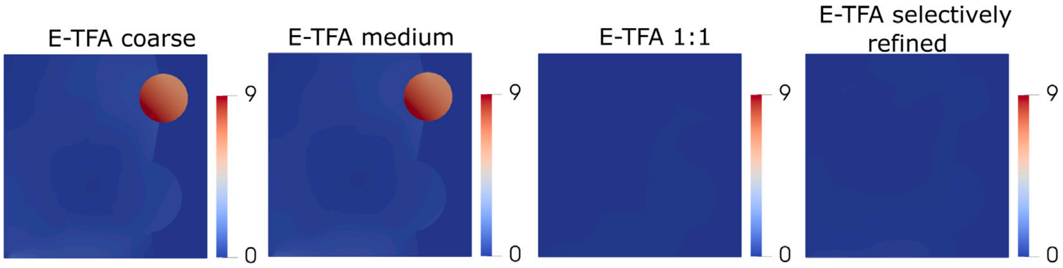


Fig. 8. Normalized difference $\frac{\|\boldsymbol{u}_{E-TFA}-\boldsymbol{u}_{FE}\|}{\|\boldsymbol{u}_{FE}\|}$ for the biaxial test on specimen with four inclusions at load step C (highlighted in Fig. 5(a)).

Table 4
Speed-up achieved using E-TFA 1:1 subdivision with tolerance $tol = \ell/100$ for the biaxial test on a specimen with four inclusions.

FE tolerance	$tol = \ell/100$	$tol = \ell/10$	$tol = \ell$
Speed-up	∼11×	∼6×	∼3×

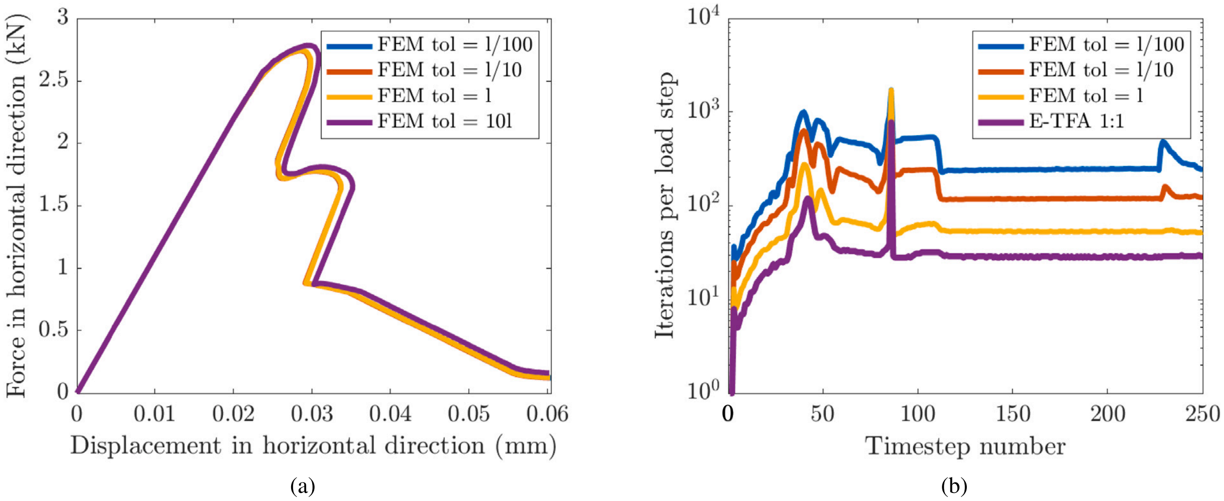


Fig. 9. (a) Load–displacement plots and (b) number of iterations required per time step for different FE analyses with varying tolerances for the biaxial test on a specimen with four inclusions.

hence, need a finer subdivision. In absence of a reference FE load–displacement curve, this strategy can be adopted to obtain an optimized subdivision. For instance, a very coarse subdivision can be selectively and recursively refined until the difference between two subsequent load–displacement curves is below a predefined tolerance.

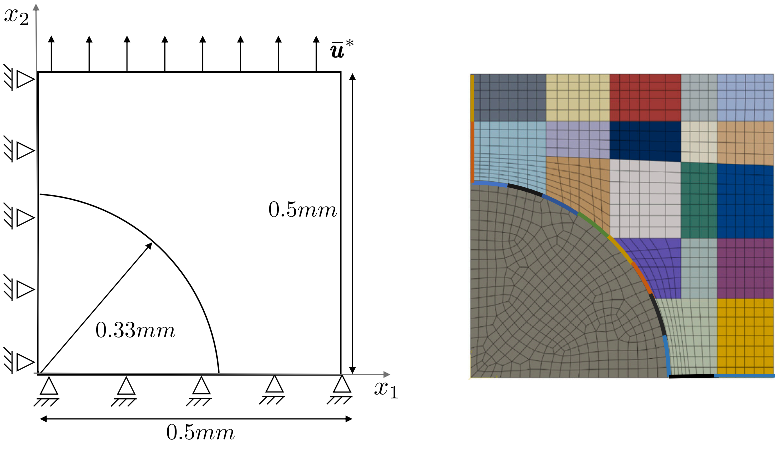
As mentioned in Section 5.1, the pre-defined tolerance values can influence the E-TFA speed-up. To analyze this effect, we perform the FE analysis using different tolerance values while keeping the one for E-TFA fixed to $tol = \ell/100$. Two additional tolerance values that provide a converged FE solution for the present example are identified to be $tol = \ell/10$ and $tol = \ell$ through successive reduction (Fig. 9(a) demonstrates that $tol = 10\ell$ is not enough to ensure a converged solution). While these tolerances give identical load–displacement plots (Fig. 9(a)), the number of iterations required per load step is reduced (Fig. 9(b)) and this influences the E-TFA speed-up. We demonstrate this using the E-TFA 1:1 (i.e. most expensive) subdivision. It can be observed from Table 4 that, with the relaxation of the FE tolerance, the E-TFA speed-up is reduced. However, even for the most relaxed case, E-TFA still takes less iterations to convergence and achieves a speed-up of about 3 times. This further corroborates the observation that E-TFA allows for a fast and robust implementation of path-following solution techniques.

5.3. Elasto-plastic matrix with a single inclusion

We now study a bi-symmetric square domain made of an elasto-plastic matrix containing a single linearly elastic circular inclusion as in [13]. Exploiting symmetry, we only consider a quarter of the domain. The inclusion is connected to the matrix by means of cohesive interfaces and decohesion of the matrix is also possible along the axes of symmetry. The geometry and boundary conditions are illustrated in Fig. 10(a), while the material parameters adopted for both matrix and interfaces are summarized in Table 5. The FE

Table 5
Material parameters for the square specimen with a single inclusion.

Parameter	Bulk		Interfaces	
	Inclusion	Matrix	Matrix-inclusion	Matrix–matrix
E (GPa)	410	75	–	–
ν (–)	0.19	0.33	–	–
σ_{y0} (MPa)	–	426	–	–
H_p (GPa)	–	2.894	–	–
$\bar{t}_T$ (MPa)	–	–	400	400
$\bar{t}_N$ (MPa)	–	–	400	400
$\bar{G}_{c,T}$ (N/mm)	–	–	1.234	12.340
$\bar{G}_{c,N}$ (N/mm)	–	–	1.234	12.340
k_T (GPa/mm)	–	–	15 000	15 000
k_N (GPa/mm)	–	–	15 000	15 000
μ (–)	–	–	0.5	0.5



(a) (b)

Fig. 10. (a) Geometry and boundary conditions of the square specimen with a single inclusion and (b) background mesh and sub-domain and sub-interface subdivision for the coarse E-TFA discretization (different colors identify different sub-domains and sub-interfaces). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Details of the numerical computations for the square specimen with a single inclusion.

	FE	E-TFA	
		Coarse	Fine
Bulk Interfaces	1126 elements 60 elements	19 sub-domains 12 sub-interfaces	31 sub-domains 24 sub-interfaces
CPU time	100% online	98.5% online 1.5% offline	97.8% online 2.2% offline
Speed-up	1 (Reference)	~6×	~5×

discretization includes 1126 quadrilateral finite elements and 60 interface elements (Fig. 10(b)). With E-TFA, we adopt two different domain and interface subdivisions (i.e., coarse and fine) whose details are given in Table 6. The fine subdivision is obtained by refining the coarse subdivision depicted in Fig. 10(b). No arc-length strategy is adopted here since no snap-back is expected. The displacement increment in vertical direction applied at each load step is $\bar{u}^* = 10^{-4}$ mm (Fig. 10(a))

Fig. 11 shows the comparison between the curves relating reaction force and imposed displacement in vertical direction obtained from the reference FE solution and the two E-TFA subdivisions. Once again a refinement in the E-TFA subdivision leads to a closer agreement with the reference solution, with the fine subdivision showing deviations from the reference curve only in the very late stage of the test. This is also confirmed by the discrepancy between E-TFA and FE in terms of area below the load–displacement curve of Fig. 11 reported in Table 7, which decreases as the number of sub-domains and sub-interfaces increases. Fig. 12 depicts the vertical displacement field at the last load step (imposed vertical displacement of $2{\cdot}10^{-2}$ mm). Here we can see that the failure mechanism predicted by the FE analysis is the decohesion of the linear elastic inclusion (Fig. 12(a)), which is well reproduced by both E-TFA analyses (Fig. 12(b)-c). Also the peak load is satisfactorily predicted by both E-TFA analyses with errors that remain

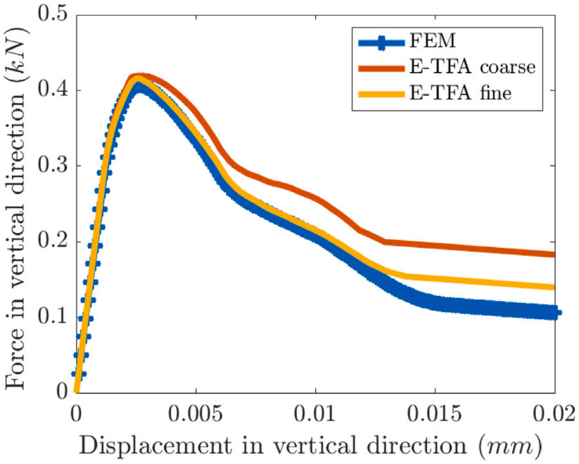
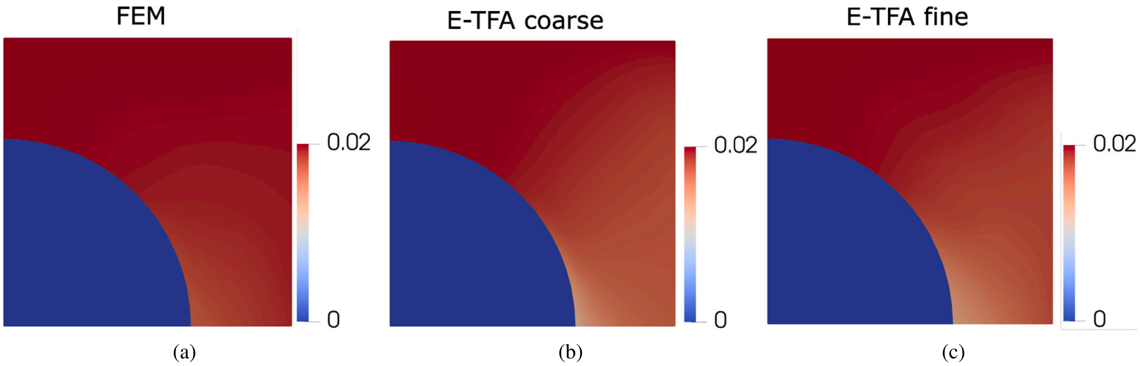


Fig. 11. Load–displacement curves for the square specimen with a single inclusion.



(a) (b) (c)

Fig. 12. Vertical displacement field for the square specimen with a single inclusion at the last load step from (a) FE and E-TFA with (b) coarse and (c) fine subdivisions.

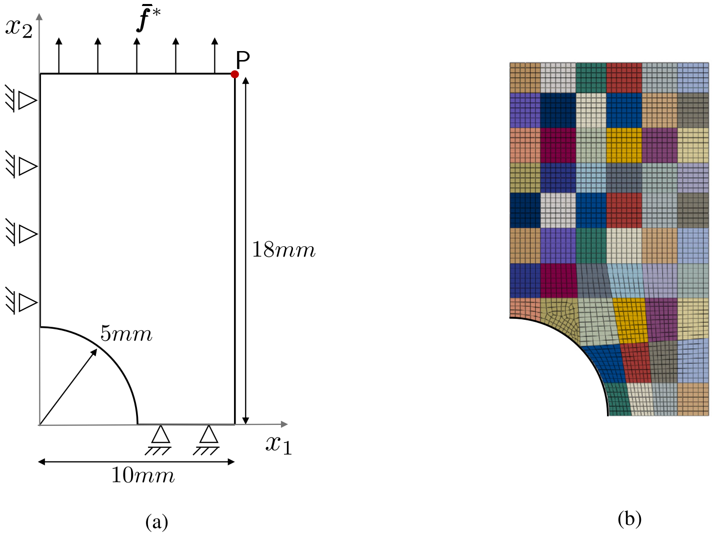
Table 7
Comparison in accuracy between FE and E-TFA global results for the square specimen with a single inclusion.

	FE	E-TFA	
		Coarse	Fine
Error in area below the load–displacement curve	– (Reference)	~21%	~9.5%
Error in peak load	– (Reference)	~5%	~4%

around 5% (Table 7 and Fig. 11). Thus, if the goal of the computation is the evaluation of the peak load and of the corresponding failure mechanism, coarse subdivisions can be adopted to maximize the gain in computational time over FE simulations.

The improvement in the accuracy of the load–displacement curve for fine subdivisions is associated to an increase in computational cost; nevertheless, E-TFA remains faster than FE even for the fine subdivision (Table 6). Since the offline CPU time is much shorter than the online time (Table 6), the reduction of the computational efficiency for fine subdivisions is due to the increase in size of the (full and unsymmetric) tangent stiffness matrix.

In Section 5.2 we demonstrate the influence of the pre-defined tolerance on the E-TFA speed-up when using the arc-length solution technique. To analyze this influence for the Newton–Raphson solution procedure, we repeat the earlier analysis by varying the FE convergence tolerance while keeping it fixed to tol = 1e−5 for E-TFA. In addition to using tol = 1e−5, a converged FE solution is now also obtained adopting tol = 1e−4. Similarly as in Section 5.2, relaxation of the FE tolerance leads to a reduction in the E-TFA speed-up from about 5× (6×) to 3× (3.3×) for the fine (coarse) subdivision.



(a) (b)

Fig. 13. (a) Geometry and boundary conditions of the plate with a hole and (b) background mesh and sub-domain subdivision for E-TFA.

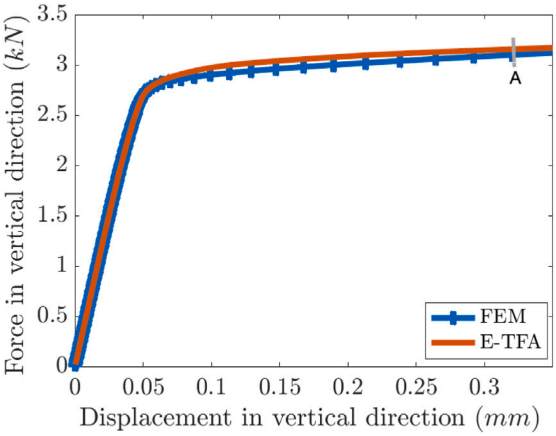


Fig. 14. Load–displacement curves for the tensile test on the plate with a hole.

Table 8
Details of the numerical computations for the tensile test on the plate with a hole.

	FE	E-TFA
Bulk	2611 elements	56 sub-domains
CPU time	100% online	80% online 20% offline
Speed-up	1 (Reference)	~4×

5.4. Plate with a hole

Next, we consider a plate with a hole subjected to tension as illustrated in Fig. 13(a). We assume an elasto-plastic material featuring isotropic hardening with Young's modulus $E = 70$ GPa, Poisson's ratio $\nu = 0.2$, initial yield stress $\sigma_{y0} = 0.243$ MPa and isotropic hardening parameter $H_p = 0.2$ MPa [39]. No interfaces are present. Exploiting symmetry, a quarter of the domain is discretized using 2611 quadrilateral finite elements, grouped for E-TFA into 56 regular sub-domains (Fig. 13(b) and Table 8). Since the response of the specimen is expected to be stable, no arc-length strategy is adopted here. The vertical force increment applied at each load step is $\bar{f}^* = 10^{-3}$ N/mm (Fig. 13(a))

Fig. 14 shows the close agreement between FE and E-TFA results in terms of total applied force vs. vertical displacement of the point P in Fig. 13(b). The limited differences in Fig. 14 are due to the coarser plastic strain discretization in E-TFA (based on the

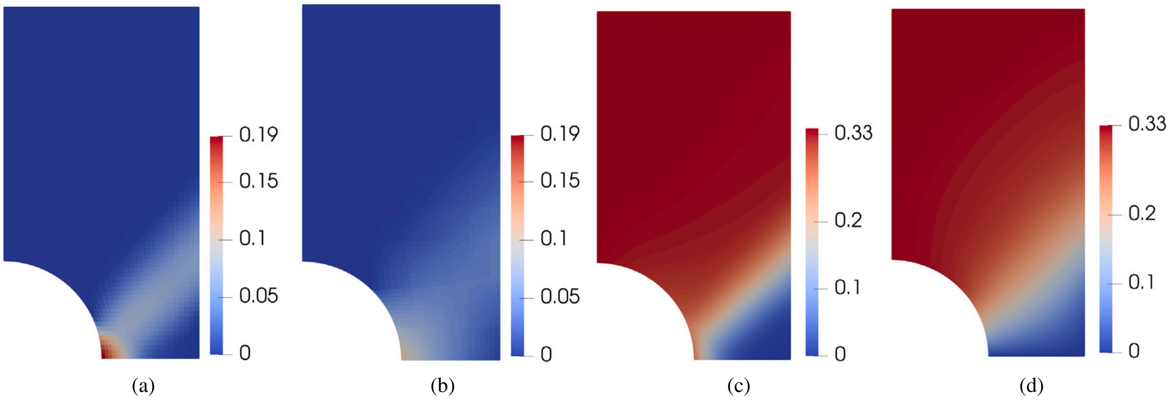


Fig. 15. Normal plastic strain in vertical direction for load step A from (a) FE and (b) E-TFA. Displacement field in vertical direction for load step A from (c) FE and (d) E-TFA.

Table 9
Material parameters for the brazilian test.

Parameter	Bulk		Interfaces	
	Aggregates	Matrix	Matrix-aggregate	Matrix–matrix
E (GPa)	150	25	–	–
ν (–)	0.2	0.2	–	–
$\bar{t}_T$ (MPa)	–	–	10	20
$\bar{t}_N$ (MPa)	–	–	5	6
$\bar{G}_{c,T}$ (N/mm)	–	–	0.3	0.6
$\bar{G}_{c,N}$ (N/mm)	–	–	0.03	0.06
k_T (GPa/mm)	–	–	1000	1000
k_N (GPa/mm)	–	–	1000	1000
μ (–)	–	–	0.8	0.8

sub-domains in Fig. 13(b)). This is clearly visible in Fig. 15a,b where the π_{22} component of the plastic strains is shown for the load step labeled A in Fig. 14. Here, the localization of the plastic strains predicted by the FE analysis in correspondence of the hole (Fig. 15a) is smoothed over multiple adjacent sub-domains (Fig. 15b). However, provided that the E-TFA discretization is not too coarse in the zone where the localization takes place [13], the effect on the displacement field is moderate (Fig. 15c,d).

E-TFA yields in this case a computational speed-up of about 4 times with most of the time taken for the online computation (Table 8). A further refinement of the E-TFA subdivision leads to the erosion of the computational gain to the point that it results no longer competitive compered to the FE reference solution. This is due to the aforementioned increase in the size of the dense tangent matrix.

5.5. Brazilian test

Next, we consider a brazilian test on a concrete specimen (Fig. 16). The matrix and aggregates are considered linearly elastic, while cohesive elements are inserted between aggregates and matrix and within the matrix as highlighted in Fig. 16 (where solid and dashed lines indicate the aggregate-matrix and the matrix–matrix interfaces, respectively). No cohesive elements are inserted within the extreme lateral parts of the specimen since they are not expected to be involved in the failure mechanism.

The material parameters are inspired by [40] and summarized in Table 9, while loading and boundary conditions are reported in Fig. 16. Two linearly elastic wooden strips with thickness of 1.35 mm, Young's modulus 10 GPa and Poisson's ratio 0.4 are included at the top and bottom of the specimen to avoid stress concentrations. This example is selected because it is particularly challenging, due to the presence of many interfaces that compete towards the final failure mechanism. Also, a sharp snap-back is expected along the equilibrium path.

The FE support mesh includes 244 interface elements, while the E-TFA analyses involve a coarse and a 1:1 subdivision (Table 10). Both FE and E-TFA analyses are performed using the arc-length approach with a reference load parameter $\bar{u}^* = 10^{-2}$ mm.

Fig. 17(a) reports the obtained load–displacement curves in vertical direction. The curve obtained from the FE analysis stops at load step A because from this point on the iterative convergence is so slow that is not possible to reach the desired tolerance within

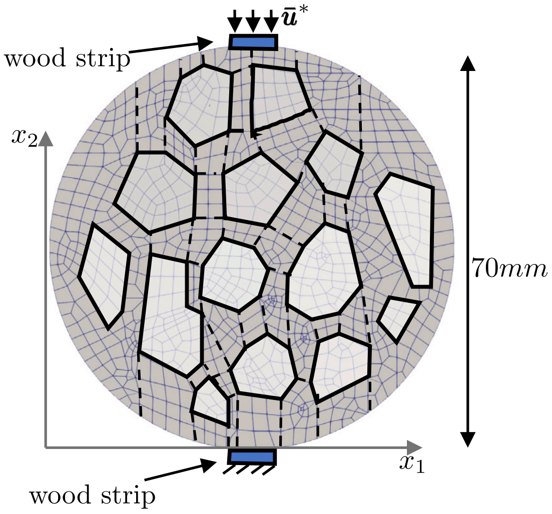


Fig. 16. Geometry and boundary conditions for the brazilian test.

Table 10
Details of the numerical computations for the brazilian test.

	FE	E-TFA	
		Coarse	1:1
Interfaces	244 elements	175 sub-interfaces	244 sub-interfaces
CPU time	100% online	99.6% online 0.4% offline	99.8% online 0.2% offline
Speed-up [(a)]	–	∼3×	1 (Reference)

[(a)] In this case the E-TFA 1:1 computation is taken as reference since the FE one has not reached convergence.

the allotted maximum number of iterations, which is set to 2000.[5] Overcoming this slow convergence rate requires the adoption of specific algorithms such as in [41–43], which is out of the scope of the present work. E-TFA, as illustrated in Fig. 17(a), is able to converge also beyond point A to the desired tolerance in less than 80 iterations (Fig. 17(b)). This is consistent with the observations in Section 5.2, and demonstrates that the implementation of the arc-length strategy within the E-TFA approach is straightforward, fast and very robust.

Knowing that E-TFA results are identical to FE results for the 1:1 subdivision, we consider the latter as reference solution. As visible in Fig. 17(a), the coarse E-TFA subdivision leads to a good compromise between computational cost and accuracy, predicting the peak load within an error of 8% and speeding-up the E-TFA computation by about 3 times with respect to the 1:1 reference. Also, in both cases the E-TFA arc-length approach is able to follow the sharp snap-back branch after the peak load and to accurately predict the failure mechanism. This is illustrated in Fig. 18 by means of the horizontal displacement field at the last load step. The result is consistent with the expected failure mechanism of an experimental brazilian test on a brittle material.

5.6. Masonry wall

With this test we study failure of a 100-mm thick masonry wall subjected to compression and shear. The details of the tests, including wall dimensions and material parameters, are inspired by [44] and reported in Fig. 19 and Table 11, respectively. The upper and lower edges of the wall are connected to two steel beams (with Young's modulus and Poisson's ratio equal to 210 GPa and 0.3, respectively) to avoid stress concentrations at the point of application of the load (Fig. 19). The mortar joints are modeled using cohesive interfaces, while the bricks are considered linearly elastic and discretized using bilinear quadrilateral elements. Since [44] identifies as one of the possible failure mechanism the split of the bricks in correspondence of the vertical joints, vertical interfaces are also placed at the center of each brick.

---

[5] The FE residual reduces steadily but slowly, hence, a higher number of allowed iteration would eventually lead to convergence. However, the computational cost would be very high.

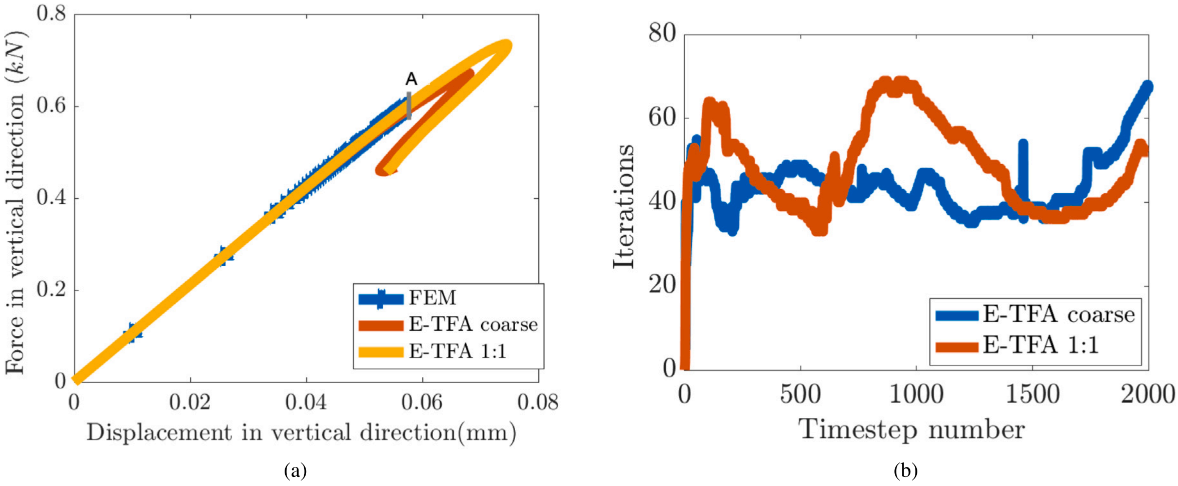


Fig. 17. (a) Load–displacement curves and (b) iterations required per time-step for different E-TFA subdivisions for the brazilian test.

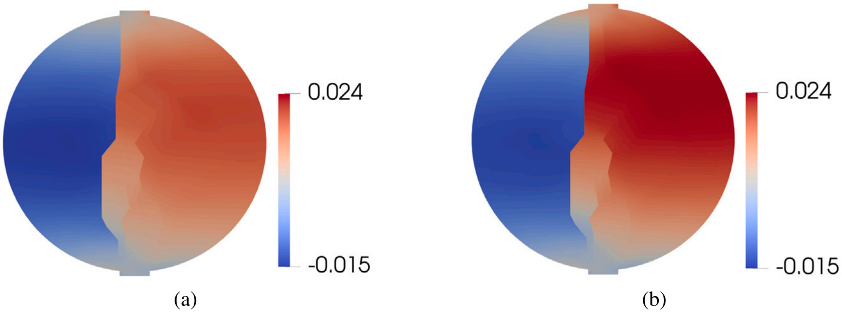


Fig. 18. Horizontal displacement field for the brazilian test at the last load step as predicted by E-TFA with (a) coarse and (b) 1:1 subdivisions.

Table 11
Material parameters for the masonry wall test.

Parameter	Bricks	Interfaces	
		Brick–mortar joint	Brick–brick
E (GPa)	16.7	–	–
ν (–)	0.15	–	–
$\bar{t}_T$ (MPa)	–	0.35	2.8
$\bar{t}_N$ (MPa)	–	0.25	2.0
$\bar{G}_{c,T}$ (N/mm)	–	0.125	0.5
$\bar{G}_{c,N}$ (N/mm)	–	0.018	0.08
k_T (MPa/mm)	–	36	461.17
k_N (MPa/mm)	–	82	1470.58
μ (–)	–	0.75	0.75

The loading procedure involves first the application of a compressive force up to a compressive vertical stress of $\sigma = 0.3$ MPa, then the horizontal displacement at the top right corner is monotonically increased until failure. In this case, no arc-length procedure is adopted and the applied displacement increment per load step is set to $\bar{u}^* = 10^{-2}$ mm (Fig. 19).

The support mesh is composed of 2332 finite elements including 868 cohesive interface elements. As shown in Section 5.2, the best trade-off between accuracy and CPU time is reached if a fine subdivision is used in the zones mostly involved by the failure mechanism while maintaining a coarser subdivision for the remain portions of the domain. The experimental evidence in [44] identifies in the interfaces highlighted in red in Fig. 19 the zones where most of the inelastic processes take place; hence, the sub-interfaces marked in red are selectively refined, while a coarser subdivision is used for the remaining interfaces (Table 12).

The force–displacement curves in Fig. 20 confirm the effectiveness of the adopted strategy: the obtained results agree well with the FE results, while ensuring a speed-up of about 2.6 times (Table 12). The agreement is not limited to global quantities, but also to the failure mechanism as shown in Fig. 21, where the horizontal displacement field is reported for the load levels A and B in Fig. 20.

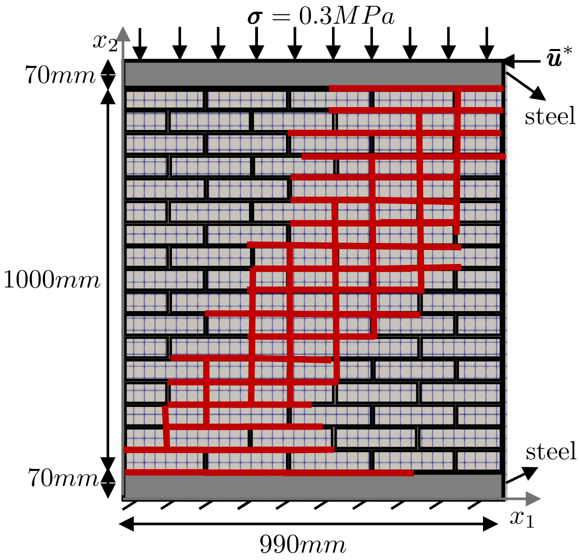


Fig. 19. Red regions indicate areas where sub-interfaces are locally refined. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

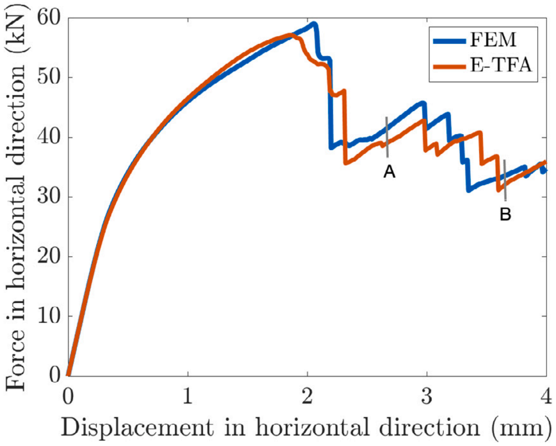


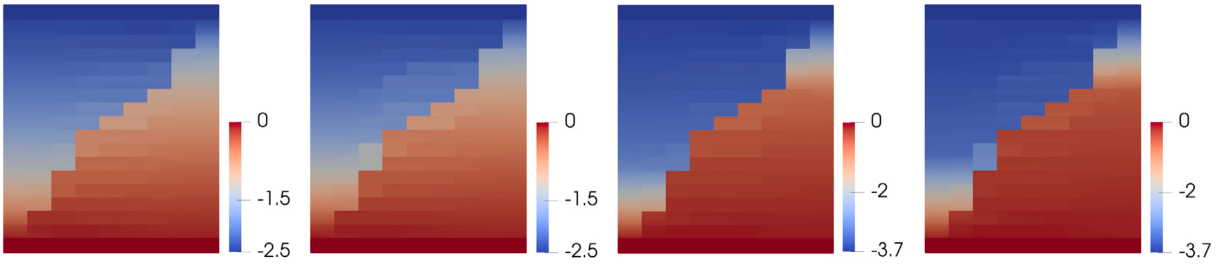
Fig. 20. Load–displacement curves for the masonry wall test.

Table 12
Details of the numerical computations for the masonry wall test.

	FE	E-TFA
Interfaces	868 elements	414 sub-interfaces
CPU time	100% online	~99.5% online ~0.5% offline
Speed-up	1 (Reference)	~2.6×

6. Conclusion

The TFA-based reduced order modeling technique originally proposed in [13,20] for multiscale analyses is extended in this paper to the single-scale setting by allowing for arbitrary Dirichlet and Neumann boundary conditions. Also, a new numerical procedure based on a consistent tangent matrix is presented, which reduces the computational cost and enables the adoption of arc-length strategies to track snap-back branches in the global response. The proposed approach, denoted as enhanced TFA (E-TFA), is able to describe the behavior of domains including cohesive interfaces to describe debonding or fracture, whereby the bulk behavior can be



(a) (b) (c) (d)

Fig. 21. Displacement field in horizontal direction at load step A from (a) FE and (b) E-TFA and at load step B from (c) FE and (d) E-TFA.

purely elastic or also inelastic. We test E-TFA by performing several numerical analyses, whose results are compared with standard FE results. The main conclusions can be summarized as follows:

- overall, E-TFA results display excellent agreement with FE results, while reducing the computational cost in most cases by approximately one order of magnitude (with the exceptions mentioned as follows);
- E-TFA results converge to FE results as the subdivision is refined. If E-TFA and FE subdivisions coincide and the same shape functions are used in both approaches, the two sets of results are identical;
- the E-TFA subdivision plays a crucial role for the achievable accuracy as well as for the computational cost. Finer subdivisions lead to a better agreement with FE results, but the computational cost increases as well due to the increment of the size of the full and unsymmetric E-TFA consistent tangent matrix;
- the potential speed-up of E-TFA also depends on whether the non-linear and inelastic behavior mostly resides in the bulk or at the interfaces. If the bulk domain exhibits inelastic behavior (in our case, plasticity), E-TFA may become barely faster than standard FE methods (at least for reasonably fine subdivisions). This is due to the larger size of the dense E-TFA consistent tangent matrix, for which operations may become less efficient than with larger but sparse system matrices in the FE method;
- the proposed arc-length approach is able to follow sharp and multiple snap-backs along the equilibrium path. Its implementation within the E-TFA technique is straightforward and very robust thanks to the fact that the constrained quantities are the main unknowns of the system. As a result, E-TFA can be significantly faster than FE even for very fine subdivisions, particularly for problems involving cohesive interfaces and an elastic bulk;
- the optimal trade-off between accuracy and computational cost is reached when finer subdivisions are located where most of the inelastic processes occur. This can be possibly achieved by adaptive refinement.

CRediT authorship contribution statement

A. Mishra: Formal analysis, Software, Visualization, Writing – original draft, Writing – review & editing, Data curation, Investigation. **P. Carrara:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing, Investigation. **S. Marfia:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **E. Sacco:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **L. De Lorenzis:** Conceptualization, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Interaction matrices and combination coefficient vectors

The three columns of each of the matrices $\boldsymbol{P}_{d,m}(\boldsymbol{x}_h)$ and $\boldsymbol{R}_{d,m}(\xi_h)$ are the solutions in terms of strains and relative displacements of the problem obtained imposing the inelastic fields $\bar{\boldsymbol{\pi}}^{(k)}(\boldsymbol{x})$ with $k = 1, 2, 3$

$$\begin{aligned}
&\text{1st column:} \quad \hat{\boldsymbol{\pi}}^{(1)}_{d,m} = \left|\begin{matrix}1\\0\\0\end{matrix}\right| \quad \Rightarrow \quad \bar{\boldsymbol{\pi}}^{(1)}(\boldsymbol{x}) = \boldsymbol{B}_{d,m}(\boldsymbol{x})\hat{\boldsymbol{\pi}}^{(1)}_{d,m}\,,\\
&\text{2nd column:} \quad \hat{\boldsymbol{\pi}}^{(2)}_{d,m} = \left|\begin{matrix}0\\1\\0\end{matrix}\right| \quad \Rightarrow \quad \bar{\boldsymbol{\pi}}^{(2)}(\boldsymbol{x}) = \boldsymbol{B}_{d,m}(\boldsymbol{x})\hat{\boldsymbol{\pi}}^{(2)}_{d,m}\,,\\
&\text{3rd column:} \quad \hat{\boldsymbol{\pi}}^{(3)}_{d,m} = \left|\begin{matrix}0\\0\\1\end{matrix}\right| \quad \Rightarrow \quad \bar{\boldsymbol{\pi}}^{(3)}(\boldsymbol{x}) = \boldsymbol{B}_{d,m}(\boldsymbol{x})\hat{\boldsymbol{\pi}}^{(3)}_{d,m}\,.
\end{aligned} \tag{A.1}$$

Following the same strategy, the two columns of the matrices $\boldsymbol{S}_{i,m}(\boldsymbol{x}_h)$ and $\boldsymbol{Q}_{i,m}(\xi_h)$ are obtained as the solution in terms of strains and relative displacements of the problem obtained imposing the inelastic relative displacements $\bar{\boldsymbol{q}}^{(k)}(\boldsymbol{x})$ with $k = 1, 2$

$$\begin{aligned}
&\text{1st column:} \quad \hat{\boldsymbol{q}}^{(1)}_{i,m} = \left|\begin{matrix}1\\0\end{matrix}\right| \quad \Rightarrow \quad \bar{\boldsymbol{q}}^{(1)}(\xi) = \boldsymbol{A}_{i,m}(\xi)\hat{\boldsymbol{q}}^{(1)}_{i,m}\,,\\
&\text{2nd column:} \quad \hat{\boldsymbol{q}}^{(2)}_{i,m} = \left|\begin{matrix}0\\1\end{matrix}\right| \quad \Rightarrow \quad \bar{\boldsymbol{q}}^{(2)}(\xi) = \boldsymbol{A}_{i,m}(\xi)\hat{\boldsymbol{q}}^{(2)}_{d,m}\,.
\end{aligned} \tag{A.2}$$

However, the storage of each single interaction matrix results inefficient from the computer memory standpoint. A more efficient way to store them can be obtained grouping the combination coefficient vectors into global vectors as follows

$$\hat{\boldsymbol{\pi}} = \left|\hat{\boldsymbol{\pi}}_{1,1} \;\dots\; \hat{\boldsymbol{\pi}}_{1,M_\Omega} \;\dots\; \hat{\boldsymbol{\pi}}_{d,1} \;\dots\; \hat{\boldsymbol{\pi}}_{d,m} \;\dots\; \hat{\boldsymbol{\pi}}_{d,M_\Omega} \;\dots\; \hat{\boldsymbol{\pi}}_{D,1} \;\dots\; \hat{\boldsymbol{\pi}}_{D,M_\Omega}\right|^T, \tag{A.3a}$$

$$\hat{\boldsymbol{q}} = \left|\hat{\boldsymbol{q}}_{1,1} \;\dots\; \hat{\boldsymbol{q}}_{1,M_\mathcal{I}} \;\dots\; \hat{\boldsymbol{q}}_{i,1} \;\dots\; \hat{\boldsymbol{q}}_{i,m} \;\dots\; \hat{\boldsymbol{q}}_{i,M_\mathcal{I}} \;\dots\; \hat{\boldsymbol{q}}_{I,1} \;\dots\; \hat{\boldsymbol{q}}_{I,M_\mathcal{I}}\right|^T, \tag{A.3b}$$

where each sub-vector $\hat{\boldsymbol{\pi}}_{d,m}$ is written in Voigt's notation and, considering (16), contains the plastic strain at the mth sampling point $\boldsymbol{x}_m$ of the dth sub-domain Ω_d, namely $\hat{\boldsymbol{\pi}}_{d,m} = \boldsymbol{\pi}(\boldsymbol{x}_m)$ with $\boldsymbol{x}_m \in \Omega_d$. Similarly, $\hat{\boldsymbol{q}}_{i,m}$ contains the inelastic relative displacements at the mth sampling point ξ_m of the ith sub-interface $\mathcal{I}_i$, namely $\hat{\boldsymbol{q}}_{i,m} = \boldsymbol{q}(\xi_m)$ with $\xi_m \in \mathcal{I}_i$.

The contributions $\boldsymbol{\varepsilon}^*$ and $\boldsymbol{s}^*$ can also be written as

$$\boldsymbol{\varepsilon}^* = \left|\boldsymbol{\varepsilon}^*_1(\boldsymbol{x}_1) \dots \boldsymbol{\varepsilon}^*_1(\boldsymbol{x}_{M_\Omega}) \dots \boldsymbol{\varepsilon}^*_d(\boldsymbol{x}_1) \dots \boldsymbol{\varepsilon}^*_d(\boldsymbol{x}_m) \dots \boldsymbol{\varepsilon}^*_d(\boldsymbol{x}_{M_\Omega}) \dots \boldsymbol{\varepsilon}^*_D(\boldsymbol{x}_1) \dots \boldsymbol{\varepsilon}^*_D(\boldsymbol{x}_{M_\Omega})\right|^T, \tag{A.4a}$$

$$\boldsymbol{s}^* = \left|\boldsymbol{s}^*_1(\xi_1) \dots \boldsymbol{s}^*_1(\xi_{M_\mathcal{I}}) \dots \boldsymbol{s}^*_i(\xi_1) \dots \boldsymbol{s}^*_i(\xi_m) \dots \boldsymbol{s}^*_i(\xi_{M_\mathcal{I}}) \dots \boldsymbol{s}^*_I(\xi_1) \dots \boldsymbol{s}^*_I(\xi_{M_\mathcal{I}})\right|^T, \tag{A.4b}$$

where each sub-vector $\boldsymbol{\varepsilon}^*_d(\boldsymbol{x}_m)$ is written in Voigt's notation and contains the total strain from the elastic localization step for the sampling point $\boldsymbol{x}_m$ of the dth sub-domain Ω_d. Also, $\boldsymbol{s}^*_i(\xi_m)$ contains the total relative displacements from the elastic localization step at the mth sampling point ξ_m of the ith sub-interface $\mathcal{I}_i$.

The global localization matrices defined in Section 4.2 can be thus written as

$$\boldsymbol{U} = \begin{bmatrix} \hat{\boldsymbol{U}}_{1,1}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{U}}_{d,m}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{U}}_{D,M_\Omega}(\boldsymbol{x}_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{U}}_{1,1}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{U}}_{d,m}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{U}}_{D,M_\Omega}(\boldsymbol{x}_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{U}}_{1,1}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{U}}_{d,m}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{U}}_{D,M_\Omega}(\boldsymbol{x}_Z) \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} \hat{\boldsymbol{T}}_{1,1}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{T}}_{d,m}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{T}}_{D,M_\Omega}(\boldsymbol{x}_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{T}}_{1,1}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{T}}_{d,m}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{T}}_{D,M_\Omega}(\boldsymbol{x}_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{T}}_{1,1}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{T}}_{d,m}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{T}}_{D,M_\Omega}(\boldsymbol{x}_Z) \end{bmatrix}, \tag{A.5a}$$

$$\boldsymbol{P} = \begin{bmatrix} \hat{\boldsymbol{P}}_{1,1}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{P}}_{d,m}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{P}}_{D,M_\Omega}(\boldsymbol{x}_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{P}}_{1,1}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{P}}_{d,m}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{P}}_{D,M_\Omega}(\boldsymbol{x}_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{P}}_{1,1}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{P}}_{d,m}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{P}}_{D,M_\Omega}(\boldsymbol{x}_Z) \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} \hat{\boldsymbol{S}}_{1,1}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{S}}_{i,m}(\boldsymbol{x}_1) & \cdots & \hat{\boldsymbol{S}}_{I,M_\mathcal{I}}(\boldsymbol{x}_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{S}}_{1,1}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{S}}_{i,m}(\boldsymbol{x}_h) & \cdots & \hat{\boldsymbol{S}}_{I,M_\mathcal{I}}(\boldsymbol{x}_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{S}}_{1,1}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{S}}_{i,m}(\boldsymbol{x}_Z) & \cdots & \hat{\boldsymbol{S}}_{I,M_\mathcal{I}}(\boldsymbol{x}_Z) \end{bmatrix}, \tag{A.5b}$$

$$\boldsymbol{R} = \begin{bmatrix} \hat{\boldsymbol{R}}_{1,1}(\xi_1) & \cdots & \hat{\boldsymbol{R}}_{d,m}(\xi_1) & \cdots & \hat{\boldsymbol{R}}_{D,M_\Omega}(\xi_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{R}}_{1,1}(\xi_h) & \cdots & \hat{\boldsymbol{R}}_{d,m}(\xi_h) & \cdots & \hat{\boldsymbol{R}}_{D,M_\Omega}(\xi_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{R}}_{1,1}(\xi_F) & \cdots & \hat{\boldsymbol{R}}_{d,m}(\xi_F) & \cdots & \hat{\boldsymbol{R}}_{D,M_\Omega}(\xi_F) \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \hat{\boldsymbol{Q}}_{1,1}(\xi_1) & \cdots & \hat{\boldsymbol{Q}}_{i,m}(\xi_1) & \cdots & \hat{\boldsymbol{Q}}_{I,M_\mathcal{I}}(\xi_1)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{Q}}_{1,1}(\xi_h) & \cdots & \hat{\boldsymbol{Q}}_{i,m}(\xi_h) & \cdots & \hat{\boldsymbol{Q}}_{I,M_\mathcal{I}}(\xi_h)\\ \vdots & & \vdots & & \vdots\\ \hat{\boldsymbol{Q}}_{1,1}(\xi_F) & \cdots & \hat{\boldsymbol{Q}}_{i,m}(\xi_F) & \cdots & \hat{\boldsymbol{Q}}_{I,M_\mathcal{I}}(\xi_F) \end{bmatrix}, \tag{A.5c}$$

where $Z = D\,M_\Omega$ is the total number of sampling points in the sub-domains and $F = I\,M_\mathcal{I}$ is the total number of sampling points in the sub-interfaces. Also, the sub-matrices in (A.5) can be written as

$$\hat{\boldsymbol{U}}_{d,m}(\boldsymbol{x}_h) = \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{U}_{d,m}(\boldsymbol{x}_h) \quad \hat{\boldsymbol{T}}_{i,m}(\boldsymbol{x}_h) = \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{T}_{i,m}(\boldsymbol{x}_h)\,, \tag{A.6a}$$

$$\hat{\boldsymbol{P}}_{d,m}(\boldsymbol{x}_h) = \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{P}_{d,m}(\boldsymbol{x}_h) \quad \hat{\boldsymbol{S}}_{i,m}(\boldsymbol{x}_h) = \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{S}_{i,m}(\boldsymbol{x}_h)\,, \tag{A.6b}$$

$$\hat{\boldsymbol{R}}_{d,m}(\xi_h) = \sum_{d=1}^{D}\sum_{m=1}^{M_\Omega} \boldsymbol{R}_{d,m}(\xi_h)\,, \quad \hat{\boldsymbol{Q}}_{i,m}(\xi_h) = \sum_{i=1}^{I}\sum_{m=1}^{M_\mathcal{I}} \boldsymbol{Q}_{i,m}(\xi_h)\,. \tag{A.6c}$$

Considering (12)–(13), the matrices $\boldsymbol{U}_{d,m}(\boldsymbol{x}_h)$ and $\boldsymbol{T}_{i,m}(\boldsymbol{x}_h)$ ($\boldsymbol{P}_{d,m}(\boldsymbol{x}_h)$ and $\boldsymbol{S}_{i,m}(\boldsymbol{x}_h)$) encode the contribution to the displacements (total strains) at the sampling point $\boldsymbol{x}_h$ for a unit excitation of the mth inelastic mode in the dth sub-domain and ith sub-interface, respectively. Similarly, $\boldsymbol{R}_{d,m}(\xi_h)$ and $\boldsymbol{Q}_{i,m}(\xi_h)$ give the total relative displacement at the point ξ_h for a unit excitation of the mth inelastic mode in the dth sub-domain and ith sub-interface, respectively. Hence, the matrices $\hat{\boldsymbol{U}}_{d,m}(\boldsymbol{x}_h)$, $\hat{\boldsymbol{T}}_{i,m}(\boldsymbol{x}_h)$, $\hat{\boldsymbol{P}}_{d,m}(\boldsymbol{x}_h)$, $\hat{\boldsymbol{S}}_{i,m}(\boldsymbol{x}_h)$, $\hat{\boldsymbol{R}}_{d,m}(\xi_h)$ and $\hat{\boldsymbol{Q}}_{i,m}(\xi_h)$ include all the contributions towards the displacements and total strains at point $\boldsymbol{x}_h$ or relative displacements at point ξ_h due to the excitations of all the modes in each sub-domain or interface.

Appendix B. Algorithms to implement the E-TFA method

The pseudo-codes for the implementation of the offline and online phases are detailed in Algo. 1 and 2, respectively.

Algorithm 1 Pseudo-code for the offline stage

```
Define FE mesh
Identify sub-domains $\Omega_d$ and sub-interfaces $\mathcal{I}_i$
Identify sampling points $\boldsymbol{x}_h$ and $\xi_h$ for every $\Omega_d$ and $\mathcal{I}_i$, respectively
Solve elastic localization problem to obtain $\boldsymbol{s}^*(\xi)$ and $\boldsymbol{\varepsilon}^*(\boldsymbol{x})$
for $i \le I$ do
    for $m \le M_\mathcal{I}$ do
        Solve linear elastic problem to obtain influence matrices $\boldsymbol{S}_{i,m}(\boldsymbol{x}_h)$ and $\boldsymbol{Q}_{i,m}(\xi_h)$ for sampling points
    end for
end for
for $d \le D$ do
    for $m \le M_\Omega$ do
        Solve linear elastic problem to obtain influence matrices $\boldsymbol{P}_{d,m}(\boldsymbol{x}_h)$ and $\boldsymbol{R}_{d,m}(\xi_h)$ for sampling points
    end for
end for
Compute and store matrices (A.5).
```

Algorithm 2 Pseudo-code for the online stage

```
for $k$ ≤total load steps do
    while $n$ ≤allowed iterations do
        Obtain consistent tangent stiffness matrix $\boldsymbol{K}^{k,n}$
        Solve for the corrections $\Delta\lambda^{k,n}$ and $\Delta\boldsymbol{y}^{k,n}$
        Obtain updated values $\lambda^{k,n}$ and $\boldsymbol{y}^{k,n}$
        if $\|\boldsymbol{r}^{k,n}\| \le tol$ then
            $\lambda^k = \lambda^{k,n}$ and $\boldsymbol{y}^n = \boldsymbol{y}^{k,n}$
            Exit the loop for the iteration
        else
            $n = n + 1$
            Proceed with the next iteration
        end if
    end while
    Obtain the updated inelastic fields $\boldsymbol{q}^k(\xi)$ and $\boldsymbol{\pi}^k(\boldsymbol{x})$
    Obtain $\boldsymbol{u}^k(\boldsymbol{x})$ by solving FE problem with updated inelastic fields and $\lambda^k$
end for
```

Appendix C. Derivative of inelastic relative displacement q with respect to total relative displacement s

Concerning the plastic deformation we have

$$\frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\varepsilon}} = \left[\frac{3G}{3G + H_p} - \sqrt{\frac{2}{3}} \frac{3G\dot{\zeta}}{\|\boldsymbol{\sigma}_d\|} \right] \frac{\boldsymbol{\sigma}_d}{\|\boldsymbol{\sigma}_d\|} \otimes \frac{\boldsymbol{\sigma}_d}{\|\boldsymbol{\sigma}_d\|} + \sqrt{\frac{2}{3}} \frac{3G\dot{\zeta}}{\|\boldsymbol{\sigma}_d\|} \mathbb{I}_d\,, \tag{C.1}$$

where $\mathbb{I}_d$ is the fourth order deviatoric identity tensor.

The derivative of inelastic relative displacements with respect to total relative displacements is defined as

$$\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{s}} = \frac{\partial D}{\partial \boldsymbol{s}} \otimes \begin{vmatrix} p \\ c \end{vmatrix} + D\, \frac{\partial}{\partial \boldsymbol{s}} \begin{vmatrix} p \\ c \end{vmatrix}. \tag{C.2}$$

The partial derivatives in (C.2) are defined as [45,46]

$$\frac{\partial D}{\partial \boldsymbol{s}} = \begin{cases} -\left[\dfrac{1}{s_{eq}^3(\eta - 1)} \begin{bmatrix} \bar{s}_T^{-2} & 0 \\ 0 & \bar{s}_N^{-2} \end{bmatrix} + \dfrac{\left(s_{eq} - 1\right)}{s_{eq}(\eta - 1)^2} \dfrac{2\left(\dfrac{\bar{s}_T \bar{t}_T}{\bar{G}_{c,T}} - \dfrac{\bar{s}_N \bar{t}_N}{\bar{G}_{c,N}} \right)}{\sqrt{s_T^2 + \langle s_N \rangle_+^2}} \begin{bmatrix} \langle s_N \rangle_+^2 & 0 \\ 0 & -s_T^2 \end{bmatrix} \right] \begin{vmatrix} s_T \\ \langle s_N \rangle_+ \end{vmatrix} & \text{if } D = D^* \\ \mathbf{0} & \text{otherwise}\,, \end{cases} \tag{C.3a}$$

$$\frac{\partial}{\partial \boldsymbol{s}} \begin{vmatrix} p \\ c \end{vmatrix} = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \text{if } s_N > 0 \\ \mathbf{0} & \text{if } s_N \leq 0 \text{ and } \Phi(\boldsymbol{t}) < 0 \\ \begin{bmatrix} 1 & \dfrac{\mu k_N t_T}{k_T |t_T|} \\ 0 & 0 \end{bmatrix} & \text{otherwise}\,. \end{cases} \tag{C.3b}$$

References

[1] G.J. Dvorak, Transformation field analysis of inelastic composite materials, Proc.: Math. Phys. Sci. 437 (1900) (1992) 311–327.

[2] G.J. Dvorak, Y. Benveniste, On transformation strains and uniform fields in multiphase elastic media, Proc. R. Soc. Lond. Ser. A 437 (1900) (1992) 291–310.

[3] J. Chaboche, S. Kruch, J. Maire, T. Pottier, Towards a micromechanics based inelastic and damage modeling of composites, Int. J. Plast. 17 (4) (2001) 411–439.

[4] J. Fish, K. Shek, M. Pandheeradi, M.S. Shephard, Computational plasticity for composite structures based on mathematical homogenization: Theory and practice, Comput. Methods Appl. Mech. Engrg. 148 (1) (1997) 53–73.

[5] K. Agathos, S.P.A. Bordas, E. Chatzi, Parametrized reduced order modeling for cracked solids, Internat. J. Numer. Methods Engrg. 121 (20) (2020) 4537–4565.

[6] D. Soldner, B. Brands, R. Zabihyan, P. Steinmann, J. Mergheim, A numerical study of different projection-based model reduction techniques applied to computational homogenisation, Comput. Mech. 60 (4) (2017) 613–625.

[7] F. Fritzen, M. Leuschner, Nonlinear reduced order homogenization of materials including cohesive interfaces, Comput. Mech. 56 (1) (2015) 131–151.

[8] J. Michel, P. Suquet, Computational analysis of nonlinear composite structures using the nonuniform transformation field analysis, Comput. Methods Appl. Mech. Engrg. 193 (48) (2004) 5477–5502.

[9] J.L. Chaboche, P. Kanouté, A. Roos, On the capabilities of mean-field approaches for the description of plasticity in metal matrix composites, Int. J. Plast. 21 (7) (2005) 1409–1434.

[10] X. Ju, R. Mahnken, Y. Xu, L. Liang, W. Zhou, A nonuniform transformation field analysis for composites with strength difference effects in elastoplasticity, Int. J. Solids Struct. 228 (2021) 111103.

[11] E. Sacco, A nonlinear homogenization procedure for periodic masonry, Eur. J. Mech. A Solids 28 (2) (2009) 209–222.

[12] S. Marfia, E. Sacco, TFA-based homogenization for composites subjected to coupled damage-friction effects, Procedia Eng. 109 (Complete) (2015) 113–120.

[13] S. Marfia, E. Sacco, Computational homogenization of composites experiencing plasticity, cracking and debonding phenomena, Comput. Methods Appl. Mech. Engrg. 304 (2016) 319–341.

[14] F. Fritzen, M. Leuschner, Reduced basis hybrid computational homogenization based on a mixed incremental formulation, Comput. Methods Appl. Mech. Engrg. 260 (2013) 143–154.

[15] S. Marfia, E. Sacco, Multiscale technique for nonlinear analysis of elastoplastic and viscoplastic composites, Composites B 136 (June 2017) (2018) 241–253.

[16] G. Alaimo, F. Auricchio, S. Marfia, E. Sacco, Optimization clustering technique for PieceWise uniform transformation field analysis homogenization of viscoplastic composites, Comput. Mech. 64 (6) (2019) 1495–1516.

[17] P. Franciosi, S. Berbenni, Multi-laminate plastic-strain organization for non-uniform TFA modeling of poly-crystal regularized plastic flow, Int. J. Plast. 24 (9) (2008) 1549–1580.

[18] P. Franciosi, S. Berbenni, Heterogeneous crystal and poly-crystal plasticity modeling from a transformation field analysis within a regularized schmid law, J. Mech. Phys. Solids 55 (11) (2007) 2265–2299.

[19] P. Suquet, Effective properties of nonlinear composites, in: P. Suquet (Ed.), Continuum Micromechanics, Springer Vienna, Vienna, 1997, pp. 197–264.

[20] J. Michel, P. Suquet, Nonuniform transformation field analysis, Int. J. Solids Struct. 40 (25) (2003) 6937–6955.

[21] S. Roussette, J. Michel, P. Suquet, Nonuniform transformation field analysis of elastic–viscoplastic composites, Compos. Sci. Technol. 69 (1) (2009) 22–27.

[22] F. Fritzen, T. Böhlke, Three-dimensional finite element implementation of the nonuniform transformation field analysis, Internat. J. Numer. Methods Engrg. 84 (7) (2010) 803–829.

[23] S. Marfia, E. Sacco, Multiscale damage contact-friction model for periodic masonry walls, Comput. Methods Appl. Mech. Engrg. 205–208 (2012) 189–203.

[24] G. Alfano, E. Sacco, Combining interface damage and friction in a cohesive-zone model, Internat. J. Numer. Methods Engrg. 68 (5) (2006) 542–582.

[25] V. Sepe, S. Marfia, E. Sacco, A nonuniform TFA homogenization technique based on piecewise interpolation functions of the inelastic field, Int. J. Solids Struct. 50 (5) (2013) 725–742.

[26] T. Mura, General theory of eigenstrains, in: Micromechanics of Defects in Solids, Springer Netherlands, Dordrecht, 1987, pp. 1–73.

[27] C.V. Verhoosel, J.J. Remmers, M.A. Gutiérrez, A dissipation-based arc-length method for robust simulation of brittle and ductile failure, Internat. J. Numer. Methods Engrg. 77 (2009) 1290–1321.

[28] M. Crisfield, A fast incremental/iterative solution procedure that handles ‘‘snap-through’’, Comput. Struct. 13 (1) (1981) 55–62.

[29] E. Ramm, Strategies for tracing the nonlinear response near limit points, in: W. Wunderlich, E. Stein, K.-J. Bathe (Eds.), Nonlinear Finite Element Analysis in Structural Mechanics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1981, pp. 63–89.

[30] I. May, Y. Duan, A local arc-length procedure for strain softening, Comput. Struct. 64 (1) (1997) 297–303.

[31] R. de Borst, Computation of post-bifurcation and post-failure behavior of strain-softening solids, Comput. Struct. 25 (2) (1987) 211–224.

[32] W. Lam, C. Morely, Arc-length method for passing limit points in structural calculations, J. Struct. Eng. 118 (1992) 169–185.

[33] M. Crisfield, An arc–length method including line search and acceleration, Internat. J. Numer. Methods Engrg. 19 (1983) 1269–1289.

[34] M. Fafard, B. Massicotte, Geometrical interpretation of the arc-length method, Comput. Struct. 4 (1993) 603–615.

[35] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, D. Wells, The deal.II finite element library: Design, features, and insights, Comput. Math. Appl. (2020).

[36] G. Alfano, M.A. Crisfield, Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues, Internat. J. Numer. Methods Engrg. 50 (7) (2001) 1701–1736.

[37] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644.

[38] T.A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, ACM Trans. Math. Software 30 (2) (2004) 196–199.

[39] E.A. de Souza Neto, D. Peric, D.R. Owen, Computational methods for plasticity : theory and applications, 2008.

[40] C. López, I. Carol, A. Aguado, Meso-structural study of concrete fracture using interface elements. I: Numerical model and tensile behavior, Mater. Struct. 41 (2007) 583–599.

[41] I. Carol, C.M. López, O. Roa, Micromechanical analysis of quasi brittle materials using fracture based interface elements, Internat. J. Numer. Methods Engrg. 52 (2001) 193–215.

[42] A. Caballero, I. Carol, C.M. López, A meso-level approach to the 3D numerical analysis of cracking and fracture of concrete materials, Fatigue Fract. Eng. Mater. Struct. 29 (12) (2006) 979–991.

[43] A. Caballero, I. Carol, C.M. López, 3D meso-mechanical analysis of concrete specimens under biaxial loading, Fatigue Fract. Eng. Mater. Struct. 30 (9) (2007) 877–886.

[44] K.F. Abdulla, L.S. Cunningham, M. Gillie, Simulating masonry wall behaviour using a simplified micro-model approach, Eng. Struct. 151 (2017) 349–365.

[45] S. Marfia, E. Sacco, J. Toti, A coupled interface-body nonlocal damage model for FRP strengthening detachment, Comput. Mech. 50 (2011) 1–17.

[46] E. Sacco, J. Toti, Interface elements for the analysis of Masonry structures, Int. J. Comput. Methods Eng. Sci. Mech. 11 (2010) 354–373.