

**ORIGINAL ARTICLE**

# An efficient static solver for the lattice discrete particle model

Dongge Jia<sup>1</sup> | John C. Brigham<sup>1,2</sup> | Alessandro Fascati<sup>1</sup><sup>1</sup>Department of Civil and Environmental Engineering, University of Pittsburgh, PA, USA<sup>2</sup>Department of Bioengineering, University of Pittsburgh, PA, USA**Correspondence**

Alessandro Fascati, Department of Civil and Environmental Engineering, University of Pittsburgh, 3700 O'Hara Street, Pittsburgh, PA 15216, USA.

Email: fascati@pitt.edu

**Abstract**

The lattice discrete particle model (LDPM) has been proven to be one of the most appealing computational tools to simulate fracture in quasi-brittle materials. Despite tremendous advancements in the definition and implementation of the method, solution strategies are still limited to dynamic algorithms, resulting in prohibitive computational costs and challenges related to solution accuracy for quasi-static conditions. This study presents a novel static solver for LDPM, introducing fundamental innovation: (1) LDPM constitutive laws are modified to provide continuous response through all possible strain/stress states; (2) an adaptive arc-length method is proposed in combination with a criterion to select the sign of the iterative load factor; (3) an adaptive limit-unloading-reloading path switch algorithm is proposed to restrict oscillations in the global stiffness matrix. Extensive validation of the proposed approach is presented. Numerical results demonstrate that the static solver exhibits satisfactory convergence rates, significantly outperforming available dynamic solutions in computational efficiency.

**KEYWORDS:**

lattice discrete particle model, arc-length method, fracture mechanics, quasi-brittle materials, heterogeneous fracture

## **1 | INTRODUCTION**

The Lattice Discrete Particle Model (LDPM) is a mesoscale discrete model for simulating fracture in quasi-brittle heterogeneous materials, originally proposed by Cusatis, Mencarelli, Pelessone, and Baylot (2011); Cusatis, Pelessone, and Mencarelli (2011). By combining the strengths of the Confinement Shear Lattice (CSL) Model (Cusatis, Bažant, & Cedolin 2003a 2003b) and the Discrete Particle Model (DPM) (Pelessone 2005), LDPM has been proven capable of realistic representation of fracture processes in quasi-brittle heterogeneous materials under complex 3-D loading conditions. The discrete nature of LDPM stems from the material idealization that allows for a mesoscale description of the governing processes. This choice allows for a direct representation of the internal structure of members, eliminating sensitivity to element size

typical of most computational models, enabling LDPM to predict size effect on mechanical responses, such as those seen for concrete specimens. By casting a parallelism with the microplane theory (Bažant & Prat 1988), formal rules can be derived to evaluate the mesoscale parameters that provide equivalent macroscopic Young's modulus and Poisson's ratio. In the presence of fracture, the exact strain energy release rates can be enforced by formulating mixed mode fracture energy-based strain softening laws at the level of the lattice elements. In addition, volumetric effects can be incorporated to simulate pore collapse and compaction in the presence of volumetric compression. As a result, the response of a quasi-brittle heterogeneous material under complex three-dimensional stress can be realistically reproduced (Cusatis, Mencarelli, et al. 2011; Zhu, Pathirage, Wang, Troemner, & Cusatis 2022).

In the last decade, several applications of LDPM for simulating fracture processes in cementitious composites have been

reported in the literature, including plain (Cusatis, Mencarelli, et al. 2011; Fascetti, Bolander, & Nisticò 2018; Shen et al. 2022) and reinforced concrete (Alnaggar, Pelessone, & Cusatis 2019), high-performance concrete (Sherzer, Alghaladis, & Peterson 2022), and fiber-reinforced composites (Fascetti, Feo, Nisticò, & Penna 2016; Shen et al. 2021). Moreover, researchers have explored the ability of LDPM to simulate geomaterials (Fascetti & Oskay 2019; Li, Zhou, Carey, Frash, & Cusatis 2018), as well as predicting failure in brick/stone masonry (Mercuri, Pathirage, Gregori, & Cusatis 2023), strain-hardening cementitious composites (Zhu, Troemner, Wang, Cusatis, & Zhou 2023), granular rocks (Ashari, Buscarnera, & Cusatis 2017), ultra-high performance (Angiolilli, Gregori, Pathirage, & Cusatis 2020), and high-permeability cementitious composites (Fascetti, Ichimaru, & Bolander 2022). In addition to expanding the scope of LDPM's application, several research groups employed the method to study the degradation of concrete-like materials under different loading and environmental conditions including cyclic loading (Zhu et al. 2022), blast and projectile penetration (Cusatis, Di Luzio, & Cedolin 2011). Description of the mesoscale effects inducing fracture also enables LDPM to be employed for simulations of Alkali Silica Reaction (Alnaggar, Cusatis, & Di Luzio 2013), aging (Wan-Wendner, Wan-Wendner, & Cusatis 2018), creep (Gan et al. 2021), and shrinkage (Abdellatef et al. 2015) in concrete elements. Another direction in which research efforts have been dedicated is multiphysics LDPM frameworks to model mechanical response and mass transport (Cibelli, Ferrara, & Di Luzio 2023; Eliáš & Cusatis 2022; Eliáš, Yin, & Cusatis 2022; Li et al. 2018).

## 1.1 | Available solution strategies

Despite many fundamental advancements in the definition and implementation of different LDPM approaches for a broad range of engineering problems, the development of a stable static solver for LDPM has eluded the research community since the method was first introduced. Most of the LDPM applications described in the previous rely on a dynamic explicit algorithm, based on a central difference scheme, to solve the equation of motion of each particle in the domain. This solution method was first proposed along with the creation of LDPM in the early 2010's (Cusatis, Pelessone, & Mencarelli 2011). The main benefit of the dynamic explicit solution is that it is not affected by convergence issues associated with strain softening damage laws and damage localization. Furthermore, under the assumption of a diagonalized (i.e., lumped) mass matrix, the explicit dynamic solution does not require the computation of global matrices, and is therefore advantageous for its ease of implementation and potential for parallelization

and utilization of high-performance computing. However, several drawbacks arise from the adoption of an explicit dynamic solver when simulating quasi-static problems: (1) the requirement for specific treatment of loading rates when analyzing quasi-static conditions (imposing limits on the allowable kinetic energy of each particle in the domain), (2) the conditional stability of the integration scheme, which often results in prohibitive computational cost, (3) the accuracy of the obtained solution for problems with significant local instabilities (such as quasi-brittle fracture). Recent works proposed dynamic implicit algorithms, for tensile-dominated failures (Pathirage, Thierry, et al. 2022) or simplified LDPM constitutive laws (Eliáš & Cusatis 2022). As a result, fundamental understanding of the accuracy of the solution method is of critical importance when solving LDPM simulations. In particular, linear stability conditions may provide inaccurate estimation of the computational time steps as a result of re-hardening effects modeled in LDPM to describe the compaction of collapsed pores given high volumetric compression at the mesoscale. This can lead to unstable solutions of nonlinear problems which might lead to notable global oscillations Cusatis, Mencarelli, Pelessone, and Baylot (2010); Fascetti (2016). However, in most cases the oscillation is local and not noticeable in global curves (e.g., the arrested instability (Belytschko & Hughes 1983, section 3)). The energy balance check has been proposed as a means for detecting potential instabilities, by evaluating the force imbalance from static equilibrium. Pathirage, Thierry, et al. (2022) presented comparisons between explicit and implicit dynamic solutions of LDPM simulations of three-point bending tests. The authors concluded that the accuracy of the dynamic explicit solution is highly dependent on the choice of the computational time step, and that the magnitude of the force imbalance decreases with increasing displacement. The relatively large observed force imbalance can also leads to high-frequency oscillations in the post-peak load-displacement curve. This aspect is particularly critical when considering that the oscillations can affect the solution path as a result of bifurcation resulting from damage localization. This effect, however, can not be controlled by the dynamic solution itself, since the dynamic solution always tends to pass dynamically to the nearest stable branch. These findings indicate that the dynamic solutions of LDPM can not always guarantee a satisfactory accuracy of the mechanical response when simulating quasi-static conditions. Therefore, more research is needed to investigate the potential inaccuracies induced by the explicit dynamic solutions for complex three-dimensional loading conditions.

## 1.2 | Novelty and motivation

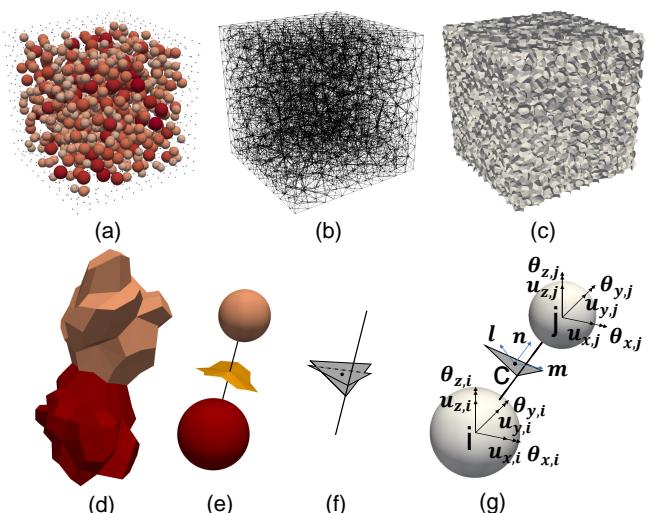
For all the aforementioned reasons, devising an efficient static solution strategy for LDPM could allow for the solution of

problems with greater complexity, while at the same time shedding light on the accuracy challenges associated with widely adopted dynamic solution schemes. In a recent work by Pathirage, Thierry, et al. (2022), preliminary investigations of a static solution of LDPM were discussed for tensile-dominated failure modes (i.e., three-point bending simulations). The complexity of the solution, however, increases significantly in presence of tensile-shear interactions, such as in uniaxial compression tests, or three-dimensional stress states. The main contribution to the field and novelty of the proposed work, therefore, is the definition, implementation and validation of an efficient and generalizable static scheme for LDPM simulations of complex failure mechanisms with interacting shear and tensile fracture. Several techniques are proposed herein to achieve the research goal, namely: (1) a local modification of the LDPM constitutive law to render the material response continuous in the entire latent space; (2) the definition and implementation of an adaptive arc-length method for the solution of the governing equations with (i) two collaborative criteria to select the initial load increment factor, employed to improve the prediction of the direction of continuation and (ii) an adaptive path switch algorithm to restrict oscillations in the global stiffness matrix caused by LDPM unloading-reloading rules (by casting a parallelism with the sequentially linear analysis used in Peridynamics (Ni, Zaccariotto, Zhu, & Galvanetto 2019)). Moreover, the numerical techniques employed to obtain computationally efficient implementation of the method are quantitatively analyzed, and recommendations are provided.

## 2 | THE LATTICE DISCRETE PARTICLE MODEL

This work follows the original procedure to create the computational domain for LDPM simulations of concrete materials, as presented in (Cusatis, Pelessone, & Mencarelli 2011; Faschetti 2016). Firstly, poly-sized spheres representing coarse aggregate particles are placed in the domain according to the given mix design ratios. To facilitate the treatment of boundary conditions, zero-radius particles are placed on the external surfaces (see Fig. 1a). Subsequently, the centroids of the aggregates are placed in the domain by means of a particle distribution algorithm. Particle distribution is implemented sequentially according to the following sequence: vertices, edges, surfaces, and interior volume. All nodes except those on the vertices of the domain are distributed through a minimum distance criterion. In this work, a grid search algorithm is implemented to efficiently distribute nodes in the computational domain by means of a volume refinement strategy for solving the N-body problem (Yip, Mohle, & Bolander 2005), which reduces the

algorithmic time and space complexities to  $O(n)$ . Once the domain has been saturated with the required number of particles, Delaunay tetrahedralization of the particle centroids (i.e., the lattice nodes) is computed to generate the lattice network (Fig. 1b). The inertial properties (i.e., mass and stiffness) to be assigned to each element are obtained by performing the LDPM tessellation for the same pointset (Figs. 1c-e). This operation creates a variable number of triangles (i.e., the LDPM facets) representing the contact areas between adjacent particle assemblies. Each facet is then projected on a plane orthogonal to the lattice element (see Fig. 1f) to avoid stress locking (Cusatis, Pelessone, & Mencarelli 2011).



**Figure 1** Construction of the LDPM computational domain: (a) random placement of poly-sized particles; (b) Delaunay tetrahedralization; (c) LDPM tessellation; (d) two cells sharing contact areas; (e) a lattice strut connecting two particles; (f) LDPM facet projection; (g) nodal degrees of freedom and centroid (point C) of the projected facet.

### 2.1 | Kinematics and degrees of freedom

Each particle assembly is assumed to undergo rigid body motion (see Fig. 1g), so that  $\mathbf{u}(\mathbf{x}) = \mathbf{u}_i + \theta_i \times (\mathbf{x} - \mathbf{x}_i) = \mathbf{A}_i(\mathbf{x})\mathbf{Q}_i$ , where  $\mathbf{A}_i(\mathbf{x})$  is the rigid body motion matrix (Cusatis, Pelessone, & Mencarelli 2011),  $\mathbf{x} = [x_1, x_2, x_3]^T$  is the position of facet center,  $\mathbf{x}_i$  is the position of node  $i$ ,  $\mathbf{u}_i^T = [u_{1i} \ u_{2i} \ u_{3i}]$  is the translational motion,  $\theta_i^T = [\theta_{1i} \ \theta_{2i} \ \theta_{3i}]$  is the rotational motion of node  $i$ , and  $\mathbf{Q}_i^T = [\mathbf{u}_i^T \ \theta_i^T]$ . Using this definition, the displacement jump at the centroid  $C$  of facet  $k$  connecting particles  $i$  and  $j$  is:

$$[\mathbf{u}_{Ck}] = \mathbf{u}_{Cj} - \mathbf{u}_{Ci} = \mathbf{A}_j(\mathbf{x}_{Ck})\mathbf{Q}_j - \mathbf{A}_i(\mathbf{x}_{Ck})\mathbf{Q}_i \quad (1)$$

A displacement jump is transformed into facet strains by dividing by the lattice element length:

$$\epsilon_{Nk} = \mathbf{n}_k^T [\mathbf{u}_{Ck}] / l_e = \mathbf{B}_N^{jk} \mathbf{Q}_j - \mathbf{B}_N^{ik} \mathbf{Q}_i \quad (2)$$

$$\epsilon_{Lk} = \mathbf{l}_k^T [\mathbf{u}_{Ck}] / l_e = \mathbf{B}_L^{jk} \mathbf{Q}_j - \mathbf{B}_L^{ik} \mathbf{Q}_i \quad (3)$$

$$\epsilon_{Mk} = \mathbf{m}_k^T [\mathbf{u}_{Ck}] / l_e = \mathbf{B}_M^{jk} \mathbf{Q}_j - \mathbf{B}_M^{ik} \mathbf{Q}_i \quad (4)$$

where  $\mathbf{n}_k = (\mathbf{x}_j - \mathbf{x}_i) / l_e$ ,  $\mathbf{m}_k$  and  $\mathbf{l}_k$  are mutually orthogonal directions in the plane perpendicular to  $\mathbf{n}_k$ ;  $\mathbf{B}_N^{pk} = (1/l_e)\mathbf{n}_k^T \mathbf{A}_p(\mathbf{x}_{Ck})$ ;  $\mathbf{B}_M^{pk} = (1/l_e)\mathbf{m}_k^T \mathbf{A}_p(\mathbf{x}_{Ck})$ ;  $\mathbf{B}_L^{pk} = (1/l_e)\mathbf{l}_k^T \mathbf{A}_p(\mathbf{x}_{Ck})$ .

The constitutive relationships are enforced on each LDPM facet,  $\sigma_k = \mathbf{F}(\epsilon_k, \xi_k)$ , where  $\sigma_k$ ,  $\epsilon_k$ , and  $\xi_k$  are vectors that collect facet stresses, strains, and internal variables, respectively. The virtual work associated with a generic facet is expressed as:

$$\delta W_k = l_e A_k \sigma_k^T \delta \epsilon_k = l_e A_k (\sigma_{Nk} \delta \epsilon_{Nk} + \sigma_{Mk} \delta \epsilon_{Mk} + \sigma_{Lk} \delta \epsilon_{Lk}) \quad (5)$$

Substituting Eqs. (2)-(4) into Eq. (5), the nodal forces at nodes  $i$  and  $j$  associated with facet  $k$  are obtained:

$$\mathbf{F}_{ik}^T = -l_e A_k (\sigma_{Nk} \mathbf{B}_N^{ik} + \sigma_{Mk} \mathbf{B}_M^{ik} + \sigma_{Lk} \mathbf{B}_L^{ik}) \quad (6)$$

$$\mathbf{F}_{jk}^T = l_e A_k (\sigma_{Nk} \mathbf{B}_N^{kj} + \sigma_{Mk} \mathbf{B}_M^{kj} + \sigma_{Lk} \mathbf{B}_L^{kj}) \quad (7)$$

Summing up the nodal forces at all nodes and equating it to the external force provides the LDPM equilibrium equations.

## 2.2 | Constitutive laws

The constitutive law used in this work primarily draws upon those proposed by Cusatis, Pelessone, and Mencarelli (2011) and Fascetti et al. (2018). The continuous stress boundary from Fascetti et al. (2018) is directly adopted; and the volumetric effect on compressive stress-strain relationship uses an equation from Cusatis, Pelessone, and Mencarelli (2011).

### 2.2.1 | Elastic behavior

In the elastic stage, a linear relationship between stresses and strains is postulated:

$$\sigma_N = E_N \epsilon_N \quad (8)$$

$$\sigma_T = E_T \epsilon_T = \alpha E_N \epsilon_T \quad (9)$$

where  $\epsilon_T = \sqrt{\epsilon_M^2 + \epsilon_L^2}$  and  $\sigma_T = \sqrt{\sigma_M^2 + \sigma_L^2}$ , which represent tangential strain and tangential stress, respectively;  $E_N = E_0$ , and  $E_0$  is the effective normal modulus (Cusatis et al. 2003b);  $\alpha = E_T/E_N$  is a mesoscale shear-normal coupling parameter, controlling the macroscopic Poisson's ratio. The following approximate relationships have been proposed to link lattice level properties and macroscopic elastic parameters  $E$  and  $v$  (Cusatis, Pelessone, & Mencarelli 2011):

$$E_0 = E/(1-2v) \Leftrightarrow E = E_0(2+3\alpha)/(4+\alpha) \quad (10)$$

$$\alpha = (1-4v)/(1+v) \Leftrightarrow v = (1-\alpha)/(4+\alpha) \quad (11)$$

### 2.2.2 | Elliptical stress boundary

Fascetti et al. (2018) proposed the use of an elliptical boundary to describe the coupling between the normal and tangential strengths. This curve is continuous in all ranges of stresses and strains (Fig. 2a):

$$\sigma_N(\sigma_N - \sigma_c - \sigma_t) - \frac{\sigma_c \sigma_t}{\sigma_s^2} \sigma_T^2 = -\sigma_c \sigma_t \quad (12)$$

where  $\sigma_t$  is the tensile strength,  $\sigma_s$  is the shear strength, and  $\sigma_c$  is the compressive stress at which pore collapse is initiated. If the current combination of normal and shear stresses violates the stress boundary, non-linear behavior is triggered. To simplify the derivations, the concepts of effective strain  $\epsilon$  and effective stress  $\sigma$  are employed (Cusatis et al. 2003b):

$$\epsilon = \sqrt{\epsilon_N^2 + \alpha \epsilon_T^2} \quad (13)$$

$$\sigma = \sqrt{\sigma_N^2 + \frac{\sigma_T^2}{\alpha}} \quad (14)$$

The coupling strain  $\omega$  is introduced as a measure of the normal-shear interaction:

$$\omega = \arctan \left( \frac{\epsilon_N}{\sqrt{\alpha} \epsilon_T} \right) \quad (15)$$

Based on such definition, the elliptical boundary enforcement can be reduced to a scalar equation (Fascetti et al. 2018):

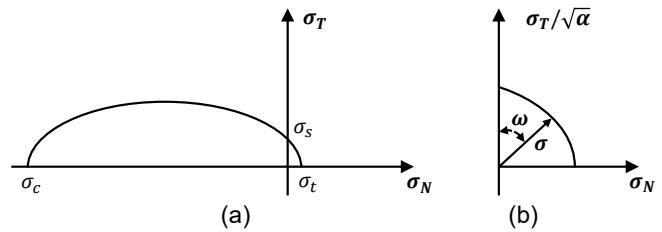
$$\sigma(\omega) = \frac{\sin(\omega)(\sigma_c + \sigma_t) + \sqrt{D}}{2(\sin^2(\omega) - \cos^2(\omega)\alpha\sigma_c\sigma_t/\sigma_s^2)} \quad (16)$$

where

$$D = \sin^2(\omega)(\sigma_c + \sigma_t)^2 - 4\sigma_c\sigma_t(\sin^2(\omega) - \cos^2(\omega)\alpha\sigma_c\sigma_t/\sigma_s^2) \quad (17)$$

Figure 2 graphically represents the elliptic boundary in the stress space. If the stresses on a certain facet do not reach the stress boundary, then Eqs. (8) and (9) apply and can be expressed by:

$$\sigma = E_0 \epsilon \quad (18)$$



**Figure 2** Elliptical stress boundary: (a)  $\sigma_N - \sigma_T$  space; and (b) graphical interpretation of the parameter  $\omega$ .

### 2.2.3 | Inelastic behavior

This Section introduces modifications to the original LDPM constitutive laws to enforce continuity for all stress-strain states, which is a fundamental component of the proposed static solver. The continuity is achieved by consolidating the original definitions of the LDPM constitutive law for the compressive and frictional behavior into a unified expression for the effective stress-strain. This also enforces the continuity in the unloading-reloading paths for all possible tangential and normal interactions, facilitating adaptive path switching.

Using the effective stress and effective strain definitions, the inelastic behavior of each strut element after the elastic stage is obtained as:

$$\sigma_b = \sigma_0 e^{-H(\epsilon_{max} - \epsilon_0)/\sigma_0} \quad (19)$$

where  $\sigma_0$  and  $\epsilon_0$  are the values of effective stress and effective strain at the onset of cracking for the current value of  $\omega$ ; and  $\epsilon_{max}$  is the maximum effective stress this element has experienced in the loading history to represent damage irreversibility. The parameter  $H$  governs the exponential post-peak behavior at the lattice element level. One of the most attractive features of LDPM is the capability of enforcing energy dissipation at the element level exactly for both tensile ( $\omega = \pi/2$ ) and shear ( $\omega = 0$ ) conditions:

$$H(\omega = \pi/2) = H_t = \frac{2E_0}{(2E_0G_t/(\sigma_t^2 l_e) - 1)} \quad (20)$$

$$H(\omega = 0) = H_s = \frac{2E_0}{(2\alpha E_0 G_s/(\sigma_s^2 l_e) - 1)} \quad (21)$$

where  $G_t$  and  $G_s$  are the prescribed values of mesoscale fracture energy for Mode I and II, respectively.

When subjected to hydrostatic compressive deformations, concrete exhibits strain-hardening plasticity due to mesoscale pore collapse. To model this effect, a dependence on volumetric strain is defined for parameter  $H$  as:

$$H(\omega = -\pi/2) = H_c = \frac{-H_{c0}}{1 + \kappa_{c2}(r_{DV} - \kappa_{c1})} \quad (22)$$

where  $r_{DV}$  is the ratio between deviatoric strain and volumetric strain,  $\epsilon_D/\epsilon_V$ ,  $\epsilon_D = \epsilon_N - \epsilon_V$ , and  $H_{c0}$ ,  $\kappa_{c1}$ , and  $\kappa_{c2}$  are material parameters. As described in Fascati et al. (2018), the volumetric strain on each lattice is obtained as a weighted average of the tetrahedral volumetric strain, weighted by the area of the LDPM facet existing on each of the tetrahedra.

Lastly, the interaction between normal and shear stresses is formulated as:

$$H = \begin{cases} H_s + (H_t - H_s)(\frac{2\omega}{\pi})^{n_t} & \text{for } \omega \geq 0 \\ H_s + (H_c - H_s)(\frac{-2\omega}{\pi})^{n_c} & \text{for } \omega < 0 \end{cases} \quad (23)$$

This constitutive law provides a continuous transition of the parameter  $H$  between tensile-, shear-, and compression-dominated stress states. The dependence of the stress-strain

curve on  $H$  for different strain states is graphically depicted in Figs. 3a and b.

The unloading-reloading rules are graphically represented in Fig. 3c. When unloading, the initial mesoscale modulus is employed until the effective stress value reaches a zero value and then remains constant for further decreases of the effective strain.

## 3 | AN EFFICIENT STATIC SOLVER FOR LDPM

The main challenge associated with the static solution of the LDPM governing equations is the existence of multiple branches in the solution, and the corresponding instability in the post-peak response of concrete structures exhibiting strain softening. This work proposes the use of a continuation method based on the parametrization of load—arc-length method. To improve convergence, adaptive arc length bisection and adaptive continuation direction are employed. Moreover, to address oscillations in the stiffness matrix caused by the LPDM unloading-reloading rules in the presence of strain softening (tension and tension/shear) and strain hardening (volumetric compression), a path switch algorithm is employed. Lastly, one of the critical steps of the solution strategy, both in terms of numerical accuracy and computational cost, is the computation of the Jacobian of the system. In this work, an automatic differentiation scheme is proposed, due to the advantage of avoiding truncation errors, in contrast to finite difference techniques. In addition, graph coloring of the Jacobian is employed to maximize the computational efficiency of the solver.

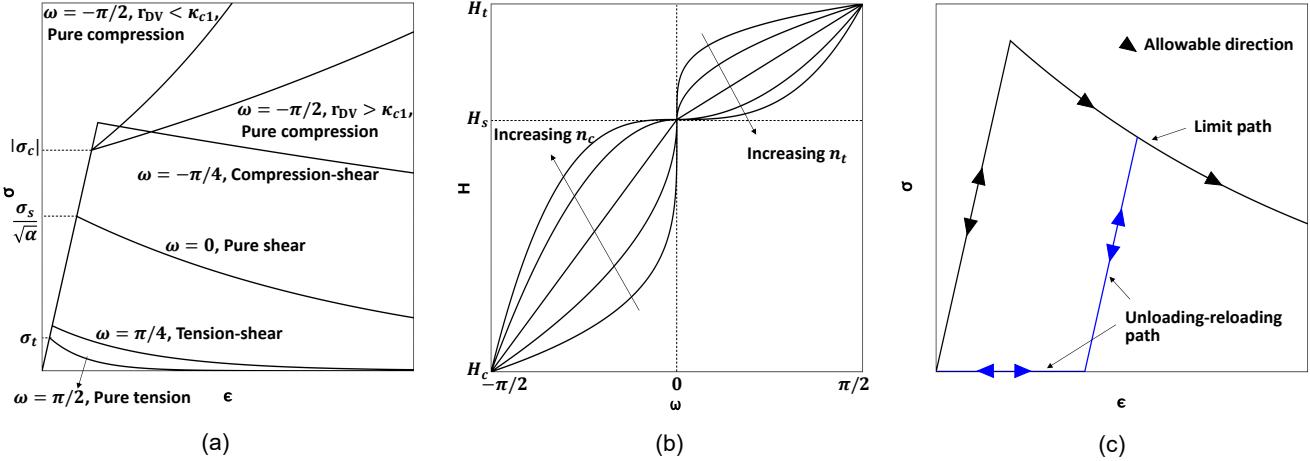
### 3.1 | Adaptive arc-length method

The arc-length method (Crisfield 1981) adopts an incremental-iterative solution procedure. In the beginning, an incremental force is imposed on the system. To get an equilibrated solution, one or more iterations are executed depending on the level of non-linearity of the system and the magnitude of the load increment:

$$\mathbf{F}^{int}(\mathbf{u}) - \mathbf{F}^{ext} = \mathbf{0} \Rightarrow \mathbf{F}^{int}(\mathbf{u}) - \lambda \mathbf{q} = \mathbf{0} \quad (24)$$

where  $\mathbf{F}^{ext}$  and  $\mathbf{F}^{int}$  are the external and the internal force vectors, respectively. Assuming that the external load has a fixed direction during the loading process,  $\mathbf{q}$  becomes a constant unit vector for the direction of the load and  $\lambda$  is its magnitude.

Let  $(\mathbf{u}_0, \lambda_0)$  be a point that satisfies equilibrium. The next point on the equilibrium path, denoted as  $(\mathbf{u}_0 + \Delta\mathbf{u}, \lambda_0 + \Delta\lambda)$ , requires computation of the variations in both  $\mathbf{u}$  and  $\lambda$ . This increases the possible solution domain in the iterations (in contrast to Newton-Raphson methods, in which either  $\Delta\lambda$  or certain components of  $\Delta\mathbf{u}$  are fixed for force- or displacement-controlled



**Figure 3** LDPM constitutive response: (a) effective stress vs. effective strain for different strain states; (b) dependence of  $H$  on coupling strain  $\omega$ ; and (c) unloading-reloading rules.

342 schemes) with the solution residual defined as:

$$\mathbf{R}(\mathbf{u}', \lambda') = \mathbf{F}^{int}(\mathbf{u}_0 + \delta\mathbf{u}) - (\lambda_0 + \delta\lambda)\mathbf{q} \quad (25)$$

343 where  $\mathbf{R}(\mathbf{u}', \lambda') = \mathbf{0}$  if and only if  $\delta\mathbf{u} = \Delta\mathbf{u}$  and  $\delta\lambda = \Delta\lambda$ .

344 Considering the non-linearity of  $\mathbf{F}^{int}$ , satisfying  $\mathbf{R}(\mathbf{u}', \lambda') = \mathbf{0}$   
345 is generally unachievable. Therefore, the first-order approximation  
346 to  $\mathbf{R}(\mathbf{u}', \lambda') = \mathbf{0}$  is obtained by Taylor series expansion:

$$\begin{aligned} \mathbf{R}(\mathbf{u}', \lambda') &= \mathbf{R}(\mathbf{u}_0, \lambda_0) + \left(\frac{\partial \mathbf{R}}{\partial \lambda}\right) \delta\lambda + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right) \delta\mathbf{u} \\ &= \mathbf{R}(\mathbf{u}_0, \lambda_0) - \mathbf{q}\delta\lambda + \mathbf{K}_T \delta\mathbf{u} \\ &= \mathbf{0} \end{aligned} \quad (26)$$

347 Repeating the first-order approximation iteratively is ex-  
348 pected to yield a sufficiently accurate approximate solution  
349 of  $\mathbf{R}(\mathbf{u}', \lambda') = \mathbf{0}$ . Equation (26) is not sufficient to determine  
350 both unknowns  $\delta\mathbf{u}$  and  $\delta\lambda$ . Thus a supplementary equation ap-  
351 pointing the arc length for the current increment ( $\Delta\mathbf{u}, \Delta\lambda$ ) is  
352 employed (i.e., the *Arc Length Equation*) (Crisfield 1981):

$$\Delta\mathbf{u}^T \Delta\mathbf{u} + \psi^2 (\Delta\lambda)^2 \mathbf{q}^T \mathbf{q} = l^2 \quad (27)$$

353 where  $\psi$  is a parameter controlling the contribution of the load  
354 increment to the length of the arc. The arc-length method re-  
355 quires Eq. (26) to be updated iteratively, while restricting the  
356 search domain by prescribing the arc length  $l$  until system equi-  
357 librium is attained. Enforcement of Eqs. (26) and (27) results  
358 in a quadratic equation for the iterative load factor  $\delta\lambda_{k,n}$ :

$$a\delta\lambda_{k,n}^2 + b\delta\lambda_{k,n} + c = 0 \quad (28)$$

359 where  $k$  represents the current increment in the loading history,  
360  $n$  is the current iteration within the increment, and

$$\begin{aligned} a &= \delta\bar{\mathbf{u}}^T \delta\bar{\mathbf{u}} + \psi^2 \mathbf{q}^T \mathbf{q} \\ b &= 2(\Delta\mathbf{u}_{k,n-1} + \delta\mathbf{u}^*)^T \delta\bar{\mathbf{u}} + 2\psi^2 \Delta\lambda_{k,n-1} \mathbf{q}^T \mathbf{q} \\ c &= (\Delta\mathbf{u}_{k,n-1} + \delta\mathbf{u}^*)^T (\Delta\mathbf{u}_{k,n-1} + \delta\mathbf{u}^*) + \psi^2 \Delta\lambda_{k,n-1}^2 \mathbf{q}^T \mathbf{q} - l^2 \end{aligned} \quad (29)$$

where

$$\begin{aligned} \delta\bar{\mathbf{u}} &\equiv \mathbf{K}_T^{-1} \mathbf{q} \\ \delta\mathbf{u}^* &\equiv -\mathbf{K}_T^{-1} \mathbf{R}_{k,n-1} \\ \delta\mathbf{u}_{k,n} &= \delta\mathbf{u}^* + \delta\lambda_{k,n} \delta\bar{\mathbf{u}} \end{aligned} \quad (30)$$

362 The quadratic equation provides two solutions. The general  
363 scheme for selecting the correct next iterational solution re-  
364 quires that the solution does not ‘‘track back’’. Therefore,  $\delta\lambda_{k,n}$   
365 is chosen in such a way that the updated  $\Delta\mathbf{u}_{k,n}$  has the minimum  
366 angle with respect to  $\Delta\mathbf{u}_{k,n-1}$ :

$$\delta\lambda_{k,n} = \arg \left[ \max_{\delta\hat{\lambda} | a\delta\hat{\lambda}^2 + b\delta\hat{\lambda} + c = 0} \{ (\Delta\mathbf{u}_{k,n-1} + \delta\mathbf{u}^* + \delta\hat{\lambda} \delta\bar{\mathbf{u}})^T \Delta\mathbf{u}_{k,n-1} \} \right] \quad (31)$$

367 The obtained value of  $\delta\lambda_{k,n}$  is then substituted into Eq. (30) to  
368 update the displacement and complete the current iteration.

369 During the first iteration of each increment (i.e.,  $n = 1$ ), Eq.  
370 (31) can not be enforced, since  $\Delta\mathbf{u}_{k,n-1}$  does not exist or is null.  
371 To resolve this issue, three separate criteria for selecting the  
372 correct  $\delta\lambda_{k,1}$  among the two solutions of Eq. (28) were proposed  
373 in the literature (de Souza Neto, Peric, & Owen 2011, p. 111):

$$\text{sign}(\delta\lambda_{k,1}) = \text{sign}(|\mathbf{K}_T(\mathbf{u}_{k-1})|) \quad (32)$$

$$\text{sign}(\delta\lambda_{k,1}) = \text{sign}(\delta\bar{\mathbf{u}}^T \mathbf{q}) \quad (33)$$

$$\text{sign}(\delta\lambda_{k,1}) = \text{sign}(\Delta\mathbf{u}_{k-1}^T \delta\bar{\mathbf{u}}) \quad (34)$$

376 However, for adoption of the arc-length method to solve LDPM  
377 simulations, none of the above can guarantee a correct choice  
378 of  $\delta\lambda_{k,1}$  that leads to at least one real solution in Eq. (28), as  
379 criteria based on the determinant of the Jacobian can fail at  
380 post-peak bifurcation points caused by the unloading-reloading  
381 paths in the constitutive model and corresponding damage lo-  
382 calization. When a bifurcation point is encountered in LDPM,  
383 two or more eigenvalues can change sign simultaneously as a  
384 result of the multiple possible co-existence of unloading and  
385 softening paths in the dense network of lattice elements. Two

or more eigenvalues changing sign together leads to an unstable sign of the determinant of the Jacobian matrix, leading Eq. (32) to fail. In contrast, Eq. (33) is not affected by such inconsistencies due to bifurcations, but experiences challenges in the “snap-back” problem (de Souza Neto et al. 2011, chapter 4). Another key reason for reduced effectiveness of all three criteria is that the determinant of the Jacobian matrix in LDPM simulations can be positive or negative in the post-peak stage. Given a known  $\Delta\mathbf{u}_{k-1}^T$  at the post-peak stage, Eq. (34) gives a sign of the wanted  $\delta\lambda_{k,1}$  that is dependent on the positive definiteness of the Jacobian, and might result in a global unloading step even in the presence of suitable positive strain increments to be selected. To overcome these issues, a novel adaptive criterion is proposed herein to choose the correct value of  $\delta\lambda_{k,1}$ , such that its sign meets the criteria:

$$\text{sign}(\delta\lambda_{k,1}) = \begin{cases} \text{sign}(\text{mean}(\delta\bar{\mathbf{u}}^l)) & \text{if } \mathbf{q}^l > \mathbf{0} \\ -\text{sign}(\text{mean}(\delta\bar{\mathbf{u}}^l)) & \text{if } \mathbf{q}^l < \mathbf{0} \end{cases} \quad (35)$$

where  $\delta\bar{\mathbf{u}}^l$  represents the components of  $\delta\bar{\mathbf{u}}$  at the nodal indices subjected to the external load,  $\mathbf{q}^l$ . The inequalities in Eq. (35) are representing element-wise operations. The logic behind the proposed criterion is to choose the correct value of  $\delta\lambda_{k,1}$ , so that the loaded nodes move in a direction consistent with the external load. The criterion guarantees a solution for all cases for which the average magnitude of the displacements at forced nodes is increasing, which means that snap-through behavior can be captured. However, snap-back behavior still poses challenges as a result of the inversion in sign of the displacement increment. Therefore, the previous criterion is supplemented by a monotonic variation of the load factor independent of displacement change:

$$\text{sign}(\delta\lambda_{k,1}) = \text{sign}(\delta\lambda_{k-1}) \quad (36)$$

Figure 4 shows an example of a global force-displacement curve with both softening and snap-back behavior. The second criterion can be used everywhere except for the limit points A and D, while the first criterion can be used in all cases except for the snap-back portion of the response (from Point B to C). By combining the two criteria, however, the adaptive arc-length method is able to capture both phenomena.

Once the values of  $\delta\lambda_{k,n}$  and  $\Delta\mathbf{u}_{k,n}$  are derived, the displacement and load fields can be updated by

$$\Delta\mathbf{u}_{k,n} = \Delta\mathbf{u}_{k,n-1} + \delta\mathbf{u}_{k,n} \quad (37)$$

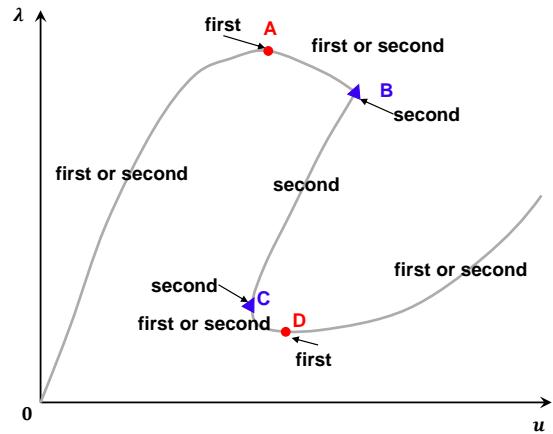
$$\Delta\lambda_{k,n} = \Delta\lambda_{k,n-1} + \delta\lambda_{k,n} \quad (38)$$

$$(\mathbf{u}_k, \lambda_k) = (\mathbf{u}_{k-1} + \Delta\mathbf{u}_{k,n}, \lambda_{k-1} + \Delta\lambda_{k,n}) \quad (39)$$

Another critical factor is the evaluation of arc length  $l$ . In the first increment, it is recommended that  $l$  is computed as:

$$l = \delta\lambda_{1,0} \sqrt{\delta\bar{\mathbf{u}}^T \delta\bar{\mathbf{u}}} \quad (40)$$

As with traditional arc-length methods, the value of  $\delta\lambda_{1,0}$  is generally assumed to be small enough to ensure convergence in



**Figure 4** Collaborative use of two criteria in the proposed adaptive arc-length method.

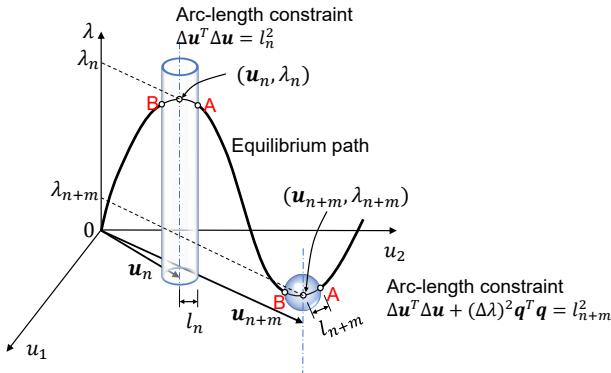
the first increment. Thus,  $\delta\lambda_{1,0}$  is required to have a value that assures the first increment is far from the nonlinear response range. For practical applications with LDPM, this is typically satisfied as long as the first increment is within 10 – 20% of the macroscopic strength of the material. After the first increment, the length of the arc in the  $k^{th}$  increment can be controlled by:

$$l_k = l_{k-1} \frac{I_d}{I_{k-1}} \quad (41)$$

where  $I_d$  is the desired number of iterations in each increment (usually  $< 5$ ) and  $I_{k-1}$  is the number of iterations required in the previous increment to achieve convergence. This equation provides smaller arc lengths when non-linearities are severe. However, the chosen arc length can still be too large for increments in which the stiffness of the system is changing significantly. Therefore, adaptive bisection of the arc length is adopted (Bellini & Chulya 1987). That is, if the arc length calculated from Eq. (41) cannot lead to a converged solution, the iterative process will restart with a bisected arc length. Such a bisection is repeated until a converged solution is obtained. If convergence still cannot be reached when the arc length is smaller than a user-defined value or a maximum number of bisections has occurred (e.g., 15 subsequent executions of bisection for a given increment), a critical point is encountered and the algorithm switches criterion (i.e., from Eq. (36) to Eq. (35) or vice versa).

When  $\psi = 1$ , the *Spherical Arc-length method* is obtained, while  $\psi = 0$  corresponds to the *Cylindrical Arc-length method*. The difference between these two constraints on the solution space is graphically described in Fig. 5. In the general cases where  $0 < \psi < 1$ , Eq. (27) represents a hyper-ellipse defined in the multidimensional displacement-load space  $(\mathbf{u}, \lambda)$ . Based on extensive numerical tests performed by the authors with LDPM, the choice of  $\psi = 0$  generally leads to faster convergence, since the magnitude of the imposed load variation is relatively

large when compared to the magnitude of the displacement variation. However, when the response is brittle, such as abrupt failures typical of tensile-splitting tests, a small increase in displacement can induce large reductions in load, and resulting in a large search space for the desired solution, which can lead to instability in the iterative process. In practice, one could estimate the value of  $\psi$  to be used based on the value of  $\Delta\lambda$  computed in the first iteration of each increment in the solution procedure, so that a preset arc-length can constrain the search space.



**Figure 5** Cylindrical and spherical arc-length methods.

Typical convergence criterion for iterative methods can be based on the norm of the force residual, the correction in primary variables (displacements herein), or variation in energy. In this work, the following convergence criterion was used for all analyses:

$$\frac{\|\mathbf{R}\|_2}{\|\mathbf{q}\|_2} \leq \epsilon_{tol} \quad (42)$$

where  $\epsilon_{tol}$  is a user-defined tolerance.

The algorithmic procedure reported below outlines the entire process to execute the proposed adaptive arc-length method in the context of solving LDPM simulations (assuming the  $n_{th}$  iteration of the  $k_{th}$  load increment):

1. If  $n = 1$ , compute the arc length of the current increment by means of Eq. (41). If  $k = 1$ , impose  $\delta\lambda_{1,0} = 1$  and compute the arc length from Eq. (40).
2. Calculate the two values of  $\delta\lambda_{k,n}$  from Eq. (28), noting that if  $n = 1$ ,  $\Delta\mathbf{u}_{k,0} = \mathbf{0}$ ,  $\delta\mathbf{u}_{k,0}^* = \mathbf{0}$  and  $\Delta\lambda_{k,0} = 0$ .
3. Use Eqs. (31), (35), or (36) to choose a value of  $\delta\lambda_{k,n}$ , then obtain  $\delta\mathbf{u}_{k,n}$  from Eq. (30), noting that if  $n = 1$ ,  $\mathbf{R}_{k,0} = \mathbf{R}_{k-1}$ .
4. Compute the displacement and load increments, and update the displacement and load fields as per Eqs. (37), (38), and (39).

5. Check if the convergence criterion is satisfied and update the increment or iteration count accordingly.

- If the L2-norm of the residual increases after one iteration or Eq. (28) has complex solutions, bisect  $l_{k-1}$ , and repeat Steps 1-4 until convergence is attained.
- If the convergence criterion cannot be satisfied after a user-defined number of bisections, impose  $k = k - \Gamma$ , switch the criterion for selecting  $\delta\lambda_{k,1}$ , and repeat Steps 1-4 until the convergence criterion is satisfied.

**Remark:** the value of  $\Gamma$  is a user-defined number of increments used to backtrack the converged solution path. When a non-convergent path is attained, a criterion switch for the evaluation of  $\delta\lambda_{k,1}$  is enforced. While doing so, stepping back several increments is recommended, since the previous selection of  $\delta\lambda_{k,1}$  may have led to several steps of nonphysical solutions before the non-convergence appeared.

### 3.2 | Incremental path switch for reducing oscillation in Jacobian

By employing the criterion described in the previous to select the value of  $\delta\lambda_{k,1}$ , the proposed adaptive arc-length method is capable of tracking the equilibrium path at the start of each increment. However, the discontinuities imposed by unloading-reloading rules in the LDPM constitutive law pose additional challenges to achieving convergence (see Fig. 6a). Such sharp changes in elemental stiffness may lead to severe instability in the Jacobian matrix between successive iterations, violating the condition for quadratic convergence of Newton-Raphson type algorithm—the Jacobian should be a sufficiently smooth function of the primary variable. To mitigate this issue, an incremental path switch strategy is proposed herein, which restricts the switch between limit and unloading-reloading paths at converged equilibrium points only. This is achieved by means of two modifications to the original LDPM constitutive model. Firstly, a parametric transition curve is defined to bridge the unloading-reloading path and the strain-softening branch, to provide continuous stiffness variations (see Fig. 6a). The slope of the transition curve at the intersection with the limit path is set to a positive value  $E_{int}$  close to zero (e.g.,  $E_N/30$ ), in consideration of the fact that the stiffness of the limit path in LDPM can both be negative (in the case of tension/shear-dominated states) and positive (in the case of compression-dominated response). Additionally, the authors performed extensive numerical investigations on this transition modulus, and found that negative values of  $E_{int}$  can lead to Hopf bifurcations (Hassard, Kazarnoff, & Wan 1981), in which the equilibrium branch is periodic. The transition curve is formulated by means of a trigonometric

function such that it is first-order continuous on the unloading-reloading path and has a positive first-order derivative at the intersection with the limit path as Eq. (43), where  $\sigma_{u-r}(\epsilon)$  is the unloading-reloading path functional form,  $\epsilon_{int}$  and  $\sigma_{int}$  are the values of the effective strain and stress at the intersection with the limit path;  $\epsilon_{start}$  is the effective strain at the start of the transition curve.

Moreover, to maximize the convergence rate, the path switch is enforced at the start of each increment, but not allowed during the iterations within the increment. After the completion of each increment, elements are checked to determine whether their constitutive paths are transitioning between limit paths and unloading-reloading paths, and the stress-strain paths will be switched accordingly. This maintains a continuous derivative throughout the following increment, without the loss of accuracy of the LDPM constitutive law.

The path switch algorithm is implemented as follows: for each lattice element in the mesh, the variation of effective strain is calculated at each converged increment, as shown in Figs. 6b and c. Two possible scenarios must be considered: (1) if an element is currently on the limit path and experiences a decrease in effective strain at a given increment, it will be identified as a transition to unloading-reloading path. Following this signal, the solution algorithm will backtrack to the previous increment, switch the path, and perform the increment again. (2) If a lattice element is currently on the unloading-reloading path and experiences an increase in the effective strain to a level greater than  $\epsilon_{int}$  but lower than  $\iota\epsilon_{int}$ , it will be identified as being in a transition back to the limit path. If the increment in effective strain is larger than  $\iota\epsilon_{int}$ , the algorithm returns to the last converged increment and bisects the arc length. As such, the transition parameter  $\iota$  provides control over the path switch. In all the simulations presented in the following, its value is taken to be  $\iota = 1.002$ , which was observed to produce negligible violations of the prescribed LDPM constitutive laws by means of the proposed path switch. This algorithm, although increasing the computational cost, provides a significant increase in the convergence rate of the solution, as demonstrated in detail in the next section.

All the previously presented modifications, including the adaptive path switch algorithm, as well as the proposed collaborative criteria, and bisection of the arc length, are graphically depicted in Fig. 7, which reports the entire algorithm proposed herein.

### 3.3 | Automatic differentiation for computing Jacobian operator

The computational efficiency and accuracy in the computation of the tangent operator  $\mathbf{K}_T$  play a crucial role in ensuring practicality and convergence of the proposed static solver. For this

reason, Automatic Differentiation (AD) is employed to derive the Jacobian operator, allowing for fast and accurate evaluation of the matrix  $\mathbf{K}_T$ . AD refers to a class of numerical techniques to compute multi-dimensional derivatives of a function specified by a computer program (Rall & Corliss 1996). The basic idea behind the method is that any given function can be described as a sequence of elementary operations and functions, so that chain rule can be applied to obtain the derivatives. Differently from numerical differentiation, AD does not introduce approximations in the computation of the derivative, and can therefore be proven to be accurate to machine precision. The elementary functions used in AD can be derived starting from the input (i.e., *forward AD*), or from the output (i.e., *reverse AD*). Forward mode computes the following recursive relation:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_j} = \frac{\partial \boldsymbol{\eta}_{,v}}{\partial \boldsymbol{\eta}_{,v-1}} \cdots \frac{\partial \boldsymbol{\eta}_{,3}}{\partial \boldsymbol{\eta}_{,2}} \frac{\partial \boldsymbol{\eta}_{,2}}{\partial \mathbf{x}_j} \quad (44)$$

while reverse mode computes the recursive relation:

$$\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}} = \frac{\partial y_i}{\partial \boldsymbol{\eta}_{,v-1}} \cdots \frac{\partial \boldsymbol{\eta}_{,3}}{\partial \boldsymbol{\eta}_{,2}} \frac{\partial \boldsymbol{\eta}_{,2}}{\partial \mathbf{x}} \quad (45)$$

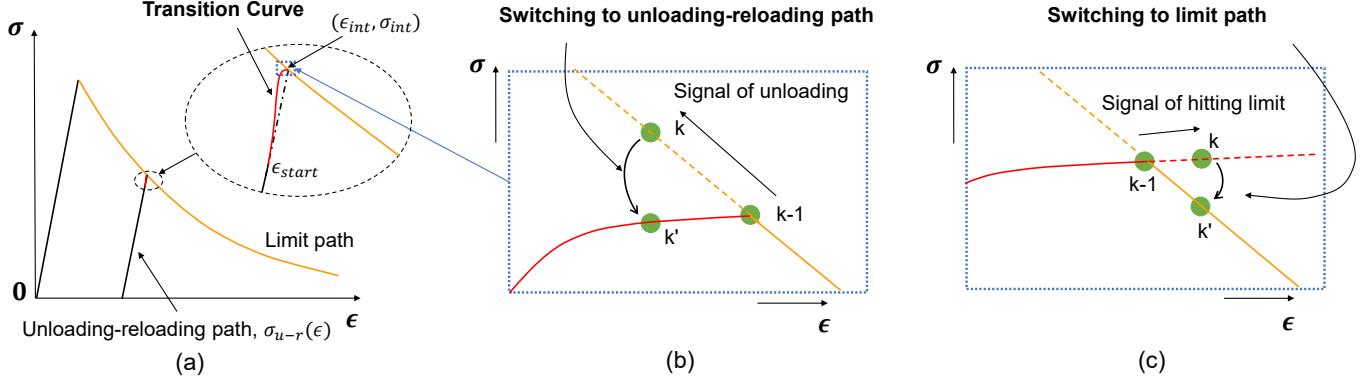
As a result of their definitions, each recursive relation in forward AD computes one column of the Jacobian operator, while each pass in reverse AD computes a row. Consequently, for functions  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  where  $n \ll m$ , forward mode generally proves to be more efficient than reverse mode, requiring  $n$  passes. The converse is true for functions where  $n \gg m$ , with reverse mode only requiring  $m$  passes. In LDPM simulations, the non-linear system of equations to be solved has approximate size  $6n_p \times 6n_p$ , where  $n_p$  is the total number of particles in the simulation (each one having 6 degrees of freedom). Therefore, forward and reverse mode AD generally yield similar computational cost. In this work, forward mode AD is utilized.

In practical implementation of AD, many sub-expressions in the automatic differentiation may be zero, independent of the input, or shared in different passes, so that their functional value can be repeatedly used in different passes. By mapping such operations, AD achieves significant computational gain when compared to finite differences, particularly in sparse systems such as the LDPM governing equations, as described in the following Section.

### 3.4 | Distance-1 graph coloring for compressing sparse Jacobian

When calculating sparse Jacobian operators, the number of required AD passes can be decreased by aggregating multiple independent passes together to allow for the computation of compressed columns (or rows) (Gebremedhin, Manne, & Pothen 2005). Figure 8 shows a structurally orthogonal matrix and two potential choices for the partition scheme. The *coloring* operation selects orthogonal columns to be compressed

$$\sigma = \begin{cases} \sigma_{u-r}(\epsilon) \left( \frac{1 - \sin\left(\frac{\epsilon - \epsilon_{start}}{\epsilon_{int} - \epsilon_{start}} \pi - \frac{\pi}{2}\right)}{2} \right) + (\sigma_{int} + E_{int}(\epsilon - \epsilon_{int})) \frac{1 + \sin\left(\frac{\epsilon - \epsilon_{start}}{\epsilon_{int} - \epsilon_{start}} \pi - \frac{\pi}{2}\right)}{2} & \text{if } \epsilon \leq \epsilon_{int} \\ \sigma_{int} + E_{int}(\epsilon - \epsilon_{int}) & \text{if } \epsilon > \epsilon_{int} \end{cases} \quad (43)$$



**Figure 6** Strategy for suppressing system stiffness oscillation: (a) transition curve; (b) switching path from the limit to unloading-reloading path; and (c) switching path from the unloading-reloading to limit path.

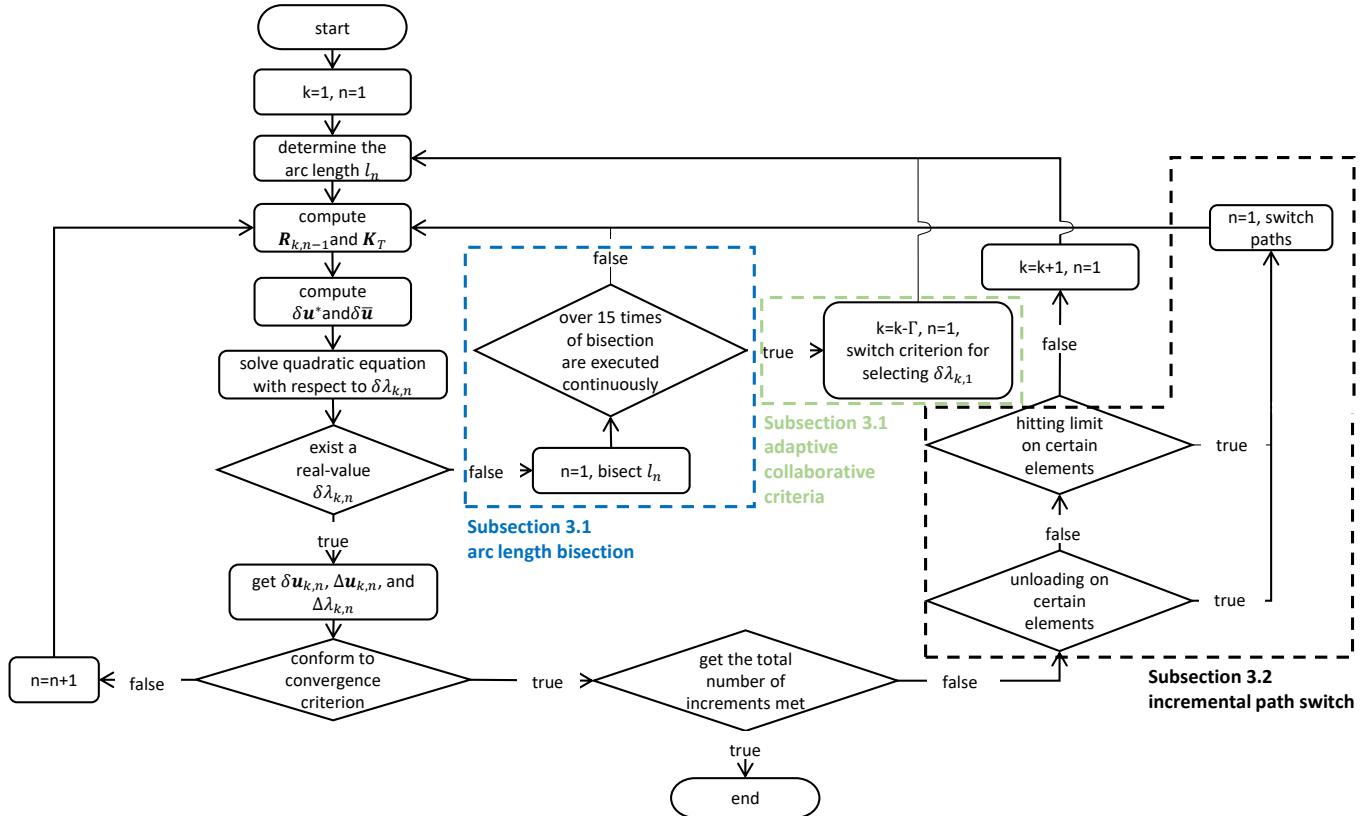
together and assigns them the same color, leading to groups of varying sized compressed operators (see Fig. 8c), and therefore different computational cost. The problem of finding the coloring scheme that produces the lowest possible number of compressed columns can be cast as a distance-1 graph coloring problem (Coleman & Moré 1983), which makes use of the column intersection graph of a matrix (with each column corresponding to a vertex in the graph, see Fig. 8d). Two vertices are connected by an edge if the corresponding columns have non-zero elements in the same row, indicating structural non-orthogonality. In such a way, a distance-1 coloring of this column intersection graph can be performed, so that adjacent vertices have different colors, effectively partitioning the original Jacobian into groups of structurally orthogonal columns. The optimization of the coloring is a known NP Complete problem (Coleman & Moré 1983). One possible solution strategy to optimize the coloring is to use a greedy algorithm, by defining the column intersection graph as  $\mathbf{G(V, E)}$ , where  $\mathbf{V}$  is the set of vertices and the edges  $\mathbf{E}$  are pairs of adjacent vertices. The algorithm assigns  $v_k$  to a sequence number of color that has already been used as much as possible. One of the issues with such an algorithm is that the final result (i.e., the number of colors and their distribution) is dependent on the numbering of the nodes. To alleviate this issue, Welsh and Powell (1967) proposed the *largest-first* (LF) ordering, also known as the Welsh-Powell Algorithm. This method organizes the vertices in a manner in which the degree  $d(v_i)$  of each vertex  $v$ —representing the number of edges connected to  $v$ —is arranged from the largest to the smallest. This ordering is strategically employed because the greedy algorithm can produce

a coloring with a maximum of  $\max\{\min[d(v_i) + 1, i] : 1 \leq i \leq n\}$  colors. Consequently, the LF ordering aims at minimizing the upper bound for the total number of colors required for partitioning the Jacobian matrix.

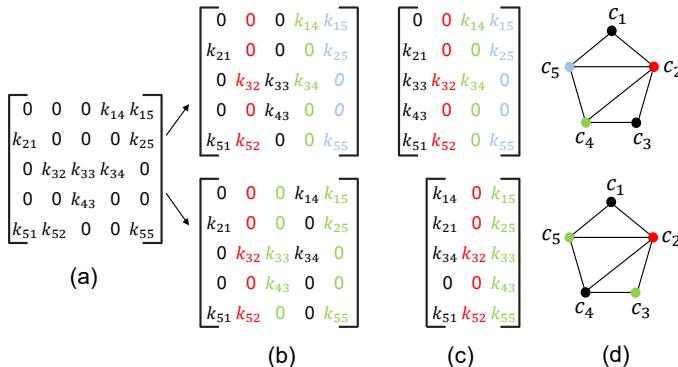
## 4 | NUMERICAL IMPLEMENTATION

Taking into account the significant computational demand associated with the solution of LDPM simulations (Lale, Rezakhani, Alnaggar, & Cusatis 2018), the proposed static solver was developed in the Julia language, which is known as one of the most competitive language for scientific computing (Bezanson, Edelman, Karpinski, & Shah 2017). This Section describes the computational techniques employed in the numerical implementation of the proposed method, and formally quantifies the computational improvement associated with their adoption.

The main objective of computationally efficient implementation of scientific computing methods in Julia is to minimize the *memory allocations* required by the algorithmic operations. First, static array vector abstract types are used to store all the element-level matrices (i.e.,  $\mathbf{B}_N$ ,  $\mathbf{B}_M$ ,  $\mathbf{B}_L$  in Eqs. (2)-(4)), avoiding superfluous heap allocations and removing loop constructs to allocate array elements directly in the CPU register for faster memory access. Second, memory allocations during access to larger matrices can be prevented by means of the `@views` macro, which yields a `SubArray` object that does not duplicate the data, but rather creates a reference to the memory segment of the original array, allowing for in-place data manipulation without necessitating memory allocation for deep copies. Lastly, in all



**Figure 7** Flowchart for the proposed adaptive arc-length method with path switch mechanism.



**Figure 8** (a) A sparse Jacobian matrix; (b) two partitions of a Jacobian matrix into groups of structurally orthogonal columns; (c) compressed Jacobian matrix after partitioning; and (d) distance-1 coloring of a column intersection graph.

the iterative operations, bounds check are prevented by means of the `@inbounds` macro, to further speed up the computation.

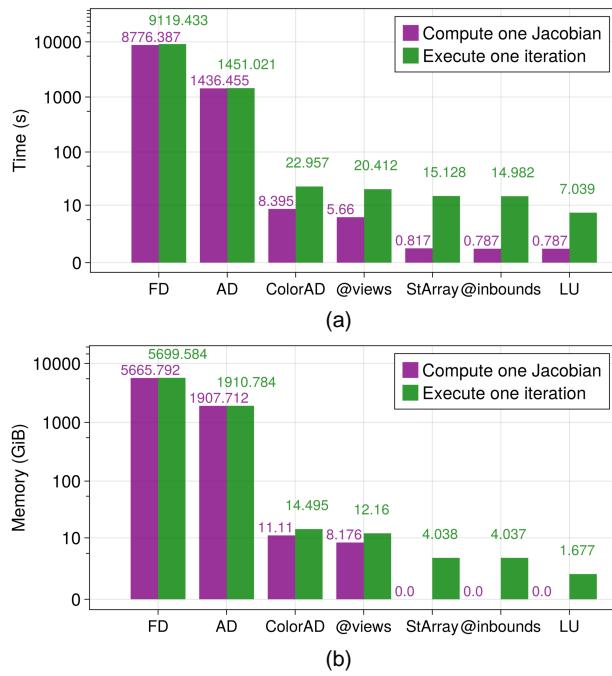
A critical point to computational mechanics methods, such as the present approach, is that the global Jacobian matrix is a highly sparse matrix with a constant sparsity pattern. This property is used in its inversion (Eq. (30)), by storing a symbolic LU factorization of the matrix through the Sparspak

library (George, Liu, & Ng 1981). This operation allows for reuse of the structure of the matrix, significantly speeding up the decomposition when solving for multiple associated linear systems.

Both the automatic differentiation of the colored Jacobian and the factorization of the resulting matrix are parallelized, and such operation is optimized to be synchronized on a single multi-threaded CPU.

#### 4.1 | Analysis of computational efficiency

This section reports the results of a set of numerical tests investigating the computational efficiency granted by the adoption of the previously described numerical techniques. All these numerical tests are performed on a workstation equipped with an Intel(R) Xeon(R) W-2245 CPU @ 3.90GHz, using the same LDPM simulation of the unconfined compression on a concrete cube as presented in Section 5.2.1. A representative point on the post-peak equilibrium path of the response curve was chosen for the execution of all the numerical tests. The corresponding system of equations yields a sparse Jacobian of size [40398 × 40398] (i.e.,  $1.63 \times 10^9$  total matrix components), with 3434854 non-zero entries.



**Figure 9** Computational efficiency improvement granted by the progressive adoption of various numerical techniques: (a) computational time and (b) memory allocation.

Figure 9 reports the total computational time and memory allocation required for the simulations. The horizontal axes show the progressive inclusion of the different numerical techniques described previously to see their effect on the computational expense, including: finite difference (FD) calculation of the Jacobian, automatic differentiation (AD), coloring-aided automatic differentiation (ColorAD), @views, StaticArray types (StArray), @inbounds, and LU factorization. An overall reduction in computing time of three orders of magnitude is observed when comparing the direct implementation of standard finite difference approximation of the Jacobian and the fully optimized algorithm. Notably, the most significant reduction in computing time is granted by the adoption of coloring of the Jacobian for automatic differentiation, which provides over two orders of magnitude reduction in cost, while not introducing any truncation error. In the numerical tests presented herein, the chromatic number is equal to  $\sim 220$  on average, which means that only around 220 passes are executed to obtain the Jacobian matrix for the global system. In contrast, AD without graph coloring would require 40398 passes. Similar results are observed in terms of total required memory allocation. In particular, no memory allocation is required in the computation of the Jacobian matrix when employing StaticArray types and color-aided AD. Also, a total reduction in memory allocation of three orders of magnitude is observed in the solution of an iterative step with the fully optimized approach.

## 5 | NUMERICAL RESULTS

The performance of the proposed LDPM static solver was evaluated by simulating quasi-static responses of concrete members under various loading conditions and comparing the estimated responses with those from corresponding experiments published in the literature. Four experiments were considered: unconfined compression (Fascetti 2016; Fascetti et al. 2018), biaxial fracture (Kupfer, Hilsdorf, & Rusch 1969), tensile splitting strength (Bazant, Kazemi, Hasegawa, & Mazars 1991), and three-point bending (Grégoire, Rojas-Solano, & Pijaudier-Cabot 2013). First, a portion of the reported test data was used to calibrate the model parameters. The model with the calibrated parameters was then used to validate the proposed solver in comparison with the remaining independent experimental results.

An important note is that all the experiments employed displacement-control protocols. However the arc-length method uses a load-controlled incremental scheme, and thus can only be employed under preset loading distributions if no special treatment is enforced. Thus, a penalty method was used to enforce the imposed boundary conditions on all the loaded and supported surfaces for the LDPM analyses.

### 5.1 | Parameters calibration

As discussed in Cusatis, Mencarelli, et al. (2011), the response of LDPM depends on both geometrical and constitutive parameters. The geometrical parameters include cement content  $c$ , water-to-cement ratio  $w/c$ , aggregate-to-cement ratio  $a/c$ , maximum aggregate size  $d_a$ , minimum aggregate size  $d_0$ , and Fuller coefficient  $n_F$ . To allow for a direct comparison with available explicit solvers for LDPM which have been extensively validated in the past, all the geometrical parameters are directly adopted from the available literature. For the specimens under unconfined compression, the LDPM geometrical parameters reported in Fascetti et al. (2018) are used. For the remaining simulations (i.e., biaxial loading, tensile splitting, and three-point bending), the LDPM geometrical parameters reported in Cusatis, Mencarelli, et al. (2011) and Pathirage, Tong, et al. (2022) are employed.

The constitutive parameters that define the mechanical behavior at each LDPM facet are closely related to the mesoscale performance of concrete. Most of the constitutive parameters in the proposed model have the same meanings as those in the original LDPM (Cusatis, Pelessone, & Mencarelli 2011), except for the following:

- Shear fracture energy,  $G_s$  governs the shear softening behavior, and consequently influences the macroscopic failure in shear- and shear-tensile dominated failures.

- Compression-shear transition exponent,  $n_c$ , governs the interaction between compression hardening and shear behavior.
- Initial hardening modulus,  $H_{c0}$ , controls the rate of variation in compressive stress, with respect to compressive strain at the onset of pore collapse.

Moreover, the current LDPM closely resembles the model proposed by Fascetti et al. (2018), with the only differences being the formulations associated with parameters  $n_c$  and  $H_{c0}$ .

These distinctions are important to consider in light of the requirements for parameter calibration, as described in the following. All the parameters that have not been reformulated when compared to the original LDPM and the model from Fascetti et al. (2018) have the same mechanical representation, therefore their values can be directly inherited from previous studies, in order to provide a direct comparison between the proposed static solver and the explicit dynamic solutions reported in the literature.

For the tests reported in Fascetti et al. (2018), the majority of the constitutive parameters were directly measured by the authors through a mesoscale testing campaign, while the tensile fracture energy  $G_t$  was calibrated in their work through best fitting of the complete load–displacement curves. As a result, all the constitutive parameters except  $n_c$  and  $H_{c0}$  were chosen to be identical in the simulations reported in the following. Furthermore, the values of  $n_c$  and  $H_{c0}$  were calibrated using a heuristic trial-and-error method, which involved adjusting these parameters to achieve the best fit between the numerical and experimental load-displacement curves of the cubic specimen.

For the biaxial fracture tests (Kupfer et al. 1969), the values of constitutive parameters have been extensively calibrated in Cusatis, Mencarelli, et al. (2011), therefore most of the values were directly obtained from the published literature. However, due to the modifications proposed herein, slight differences exist when comparing the two formulations. Therefore, for this set of simulations, calibration of parameters  $\sigma_t$ ,  $\sigma_{c0}$ , and  $n_c$  was performed on three of the reported tests (i.e., uniaxial compression, uniaxial tension, and biaxial tension), while all the remaining experimental observations were used for validation.

In both the splitting (Bazant et al. 1991) and three-point bending tests (Grégoire et al. 2013), the failure modes are tensile-dominated. This implies that only the parameters relating to tensile failure should be calibrated. All other parameters were directly adopted from Cusatis, Mencarelli, et al. (2011) and Pathirage, Tong, et al. (2022). Specifically, the values of  $\sigma_t$  and  $G_t$  were calibrated by best fit on the experimental response of notched beams subjected to three-point bending, and consequently no parameters were calibrated when modeling the tensile splitting strength tests, for which the mechanical parameters were adopted from (Cusatis, Mencarelli, et al. 2011).

**Table 1** Values of constitutive parameters used in the numerical simulations

| Symbol<br>(units)       | Unconfined<br>compression | Biaxial<br>behavior | Tensile<br>splitting | three-<br>point<br>flexure |
|-------------------------|---------------------------|---------------------|----------------------|----------------------------|
| $E_0$ (MPa)             | 44000                     | 46260               | 55610                | 57180                      |
| $\alpha$ (-)            | 0.21                      | 0.25                | 0.25                 | 0.25                       |
| $\sigma_t$ (MPa)        | 3.5                       | 2.89*               | 4.65                 | 2.62*                      |
| $G_t$ (N/mm)            | 0.04                      | 0.001               | 0.0194               | 0.0272*                    |
| $n_t$ (-)               | 1                         | 0.2                 | 0.1                  | 0.2                        |
| $\sigma_s/\sigma_t$ (-) | 10                        | 5.75                | 2.75                 | 3.276                      |
| $G_s$ (N/mm)            | 0.7                       | 0.7                 | 0.7                  | 0.7                        |
| $\sigma_{c0}$ (MPa)     | 52                        | 32.6*               | 150                  | 120                        |
| $H_{c0}/E_0$ (-)        | 0.056*                    | 0.1                 | 0.4                  | 0.4                        |
| $\kappa_{c1}$ (-)       | 1                         | 1                   | 1                    | 1                          |
| $\kappa_{c2}$ (-)       | 5                         | 5                   | 5                    | 5                          |
| $n_c$ (-)               | 2.2*                      | 0.7*                | 1                    | 1                          |

Note: superscript \* indicates calibrated values.

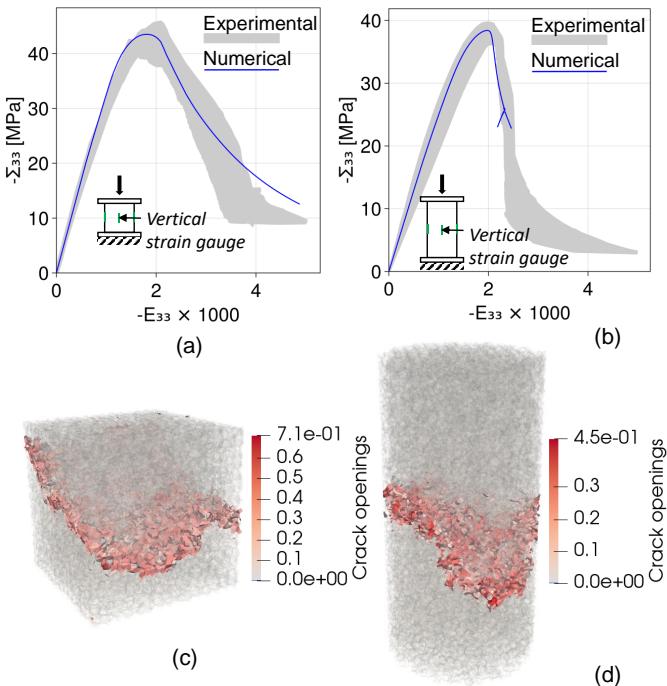
All the constitutive parameters used in the four test cases are reported in Table 1. Crack openings are also computed as shown in the following validations, by the way from Fascetti et al. (2018), that is  $\omega_c = (\omega_N^2 + \omega_T^2)^{1/2}$  and  $\omega_i = l_e(e_i - \sigma_i/E_i)$  where  $i = N, T$ .

## 5.2 | Validation

### 5.2.1 | Unconfined compression

The unconfined compression tests conducted by (Fascetti 2016; Fascetti et al. 2018) include eight specimens with different geometries. Five cubes of size  $150 \times 150 \times 150$  mm and three cylinders of size  $150 \times 300$  mm were cast and cured before testing. During the tests, the loaded ends of the cylindrical specimens were capped to prevent transverse displacement at the loaded boundary. For the cubic specimens, the load was applied through steel platens directly in contact with the top and bottom surfaces, which implies the existence of frictional forces on the boundary. According to available previous work on LDPM, a constant friction coefficient equal to 0.3 is assumed (Cusatis et al. 2003a). Frictional effects are introduced in the model by distributing the resulting force on the boundary nodes.

Figure 10a reports the comparison between numerical and experimental stress-strain curves for the cubic specimens. In the simulations, the compressive strain is measured using the virtual strain gauge technique originally proposed in (Cusatis, Pelessone, & Mencarelli 2011), according to which the local strain measured at the lattice element level is averaged in a user-defined area to mimic strain calculation in the physical



**Figure 10** Unconfined compression behavior: (a) macroscopic stress-strain curve for cube; (b) macroscopic stress-strain curve for cylinder; (c) contours of mesoscale crack opening for cube specimen; and (d) contours of mesoscale crack opening for cylindrical specimen. (Crack openings units: mm)

specimens. Figure 10b reports the comparison between numerical and experimental results on cylindrical specimens. This set of tests was used for validation purposes, and therefore is using the same set of parameters adopted in the cubic specimens simulations. The proposed model is capable of predicting the peak and post-peak responses for both cases. Moreover, the fracture patterns predicted by LDPM (see Fig. 10c and d) show excellent agreement with the experimental failures reported in Fascetti et al. (2018).

### 5.2.2 | Biaxial behavior

The applicability of the static solution of LDPM in simulating biaxial behavior of concrete was validated using classical experimental results from Kupfer et al. (1969). This set of tests was carried out on panel specimens of size  $200 \times 200 \times 50$  mm. Biaxial stress states were enforced using steel brushes, ensuring that the proportion of macroscopic stresses along two perpendicular in-plane axes,  $k = \Sigma_{33}/\Sigma_{22}$ , remained constant during the tests. Experimental evidence has shown that the friction between the steel brush and the concrete is negligible in the pre-peak stage (Vonk 1992). For this reason, since the post-peak branch recorded in the experiment is limited, frictionless

boundary conditions were utilized in the simulations without any expected loss of accuracy.

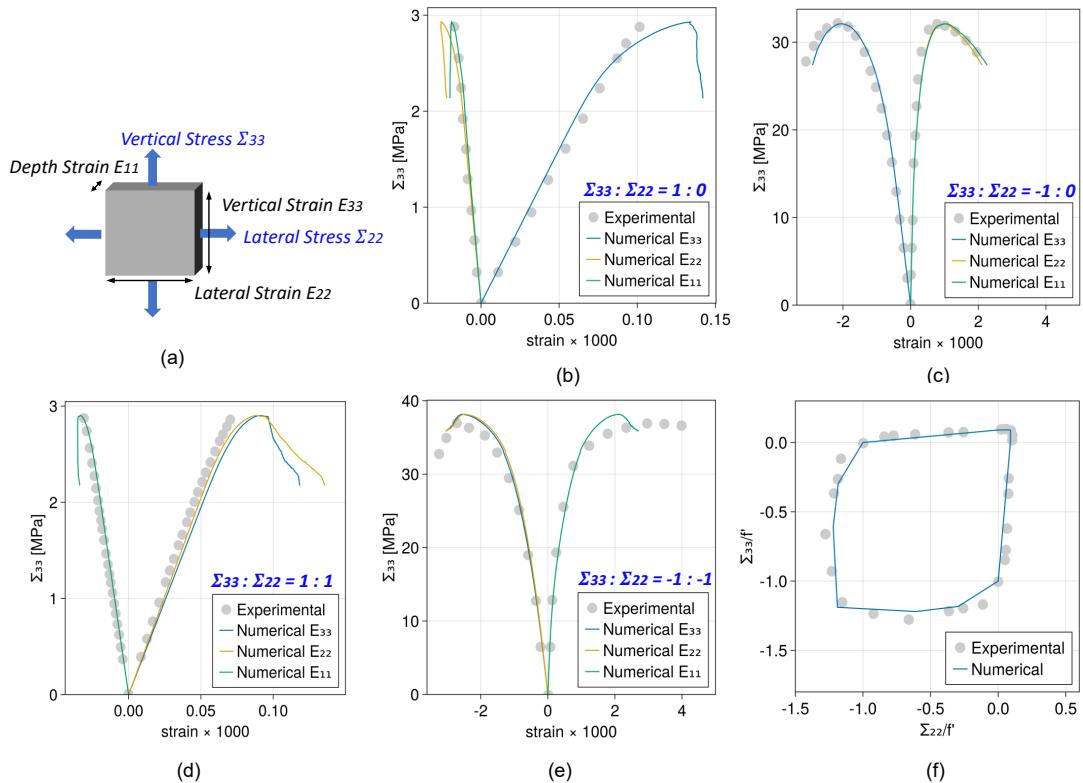
Figure 11a reports the test configuration for the biaxial fracture tests. The vertical stress  $\Sigma_{33}$  and lateral stress  $\Sigma_{22}$  are determined by dividing the vertical load and lateral load by the corresponding resisting area, respectively. The vertical strain  $E_{33}$  and lateral strain  $E_{22}$  are calculated by dividing the changes in the vertical length and lateral length by the original values. Figure 11 reports the comparison between the experimental and numerical biaxial stress-strain curves. The parameters calibrated on the uniaxial tension, uniaxial compression, and equi-biaxial tension experiments (Figs. 11b, c, and d) were used for validation of the model on the equi-biaxial compression case (see Fig. 11e), which shows excellent prediction of the peak stress values (less than 3% variation). Moreover, the full range of possible biaxial states is compared in Fig. 11f, which demonstrates excellent agreement between the experimental and numerical results.

Figure 12 reports the crack openings distribution at failure for four different biaxial stress states. For the uniaxial tension case, cracks propagate orthogonally to the applied stress. For the uniaxial compression case, failure associated with the development of diagonal shear bands is predicted by the model. Under biaxial tension and compression, the main cracks are developed on one diagonal of the specimen until failure is attained. These cracking patterns again demonstrate excellent agreement with the reported experimental results (Kupfer et al. 1969).

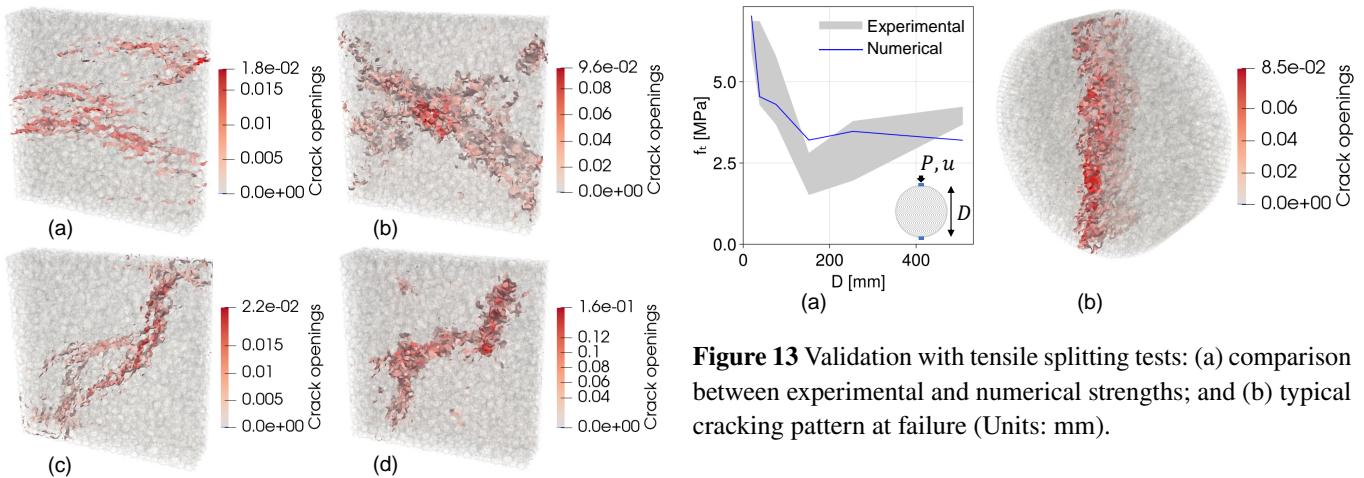
### 5.2.3 | Tensile splitting strength tests

The tests performed by Bazant et al. (1991) were used to demonstrate capability of the model to describe splitting tensile failure and associated size effect. A total of six cylindrical specimens were simulated to validate the static solution for LDPM proposed herein, with diameter values equal to 19 mm (0.75 in.), 38 mm (1.5 in.), 76 mm (3 in.), 152 mm (6 in.), 254 mm (10 in.), and 508 mm (20 in.), respectively. The thickness of all the specimens was 51 mm (2 in.). The specimens geometry and the boundary condition used in the experimental tests are shown in Fig. 13a. In the simulations, a vertical kinematic constraint was assigned to all the nodes in contact with the load distribution device, while the bottom nodes were fixed in the vertical direction.

A pronounced decreasing trend in the splitting strength and tangential modulus are observed with an increase in diameter. Figure 13a reports the comparison between the experimental and numerical strengths. The results demonstrate the capability of the proposed model to predict splitting failure (see the crack pattern at failure in Fig. 13b) and size effect in plain concrete members. Moreover, the slight deviation between experimental and numerical results for the specimens can be attributed to the

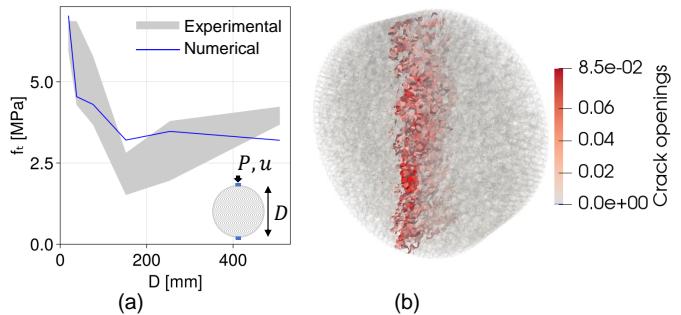


**Figure 11** Biaxial responses: (a) test configuration; macroscopic stress-strain curves for (b) uniaxial tension, (c) uniaxial compression, (d) equi-biaxial tension, (e) equi-biaxial compression; and (f) biaxial failure envelope.



**Figure 12** Biaxial cracking patterns for (a) uniaxial tension, (b) uniaxial compression, (c) equi-biaxial tension, and (d) equi-biaxial compression. (Units: mm)

local failure at the support that was attained in the experiments, as discussed by Cusatis, Mencarelli, et al. (2011).

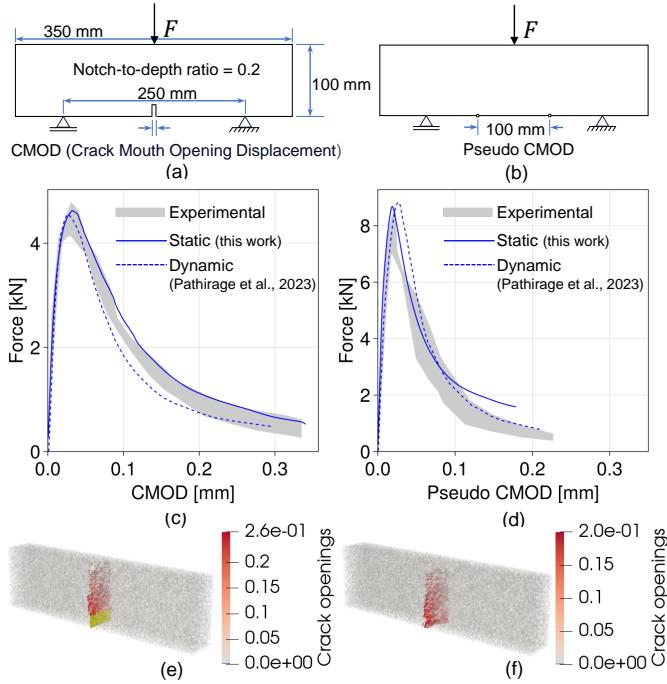


**Figure 13** Validation with tensile splitting tests: (a) comparison between experimental and numerical strengths; and (b) typical cracking pattern at failure (Units: mm).

## 5.2.4 | Three-point bending tests

As reported in Figs. 14a and b, the experimental campaign investigated the response of two sets of beams with depth of 100 mm, span-to-depth ratio of 2.5, and an out-of-plane thickness of 50 mm. The first set of beams also comprised a 2 mm wide and 20 mm long vertical notch at mid-span. The experiments were performed under crack mouth opening displacement (CMOD) control in order to obtain a stable post-peak response. The CMOD is measured by recording the distance between two

940  
941  
942  
943  
944  
945  
946  
947  
948



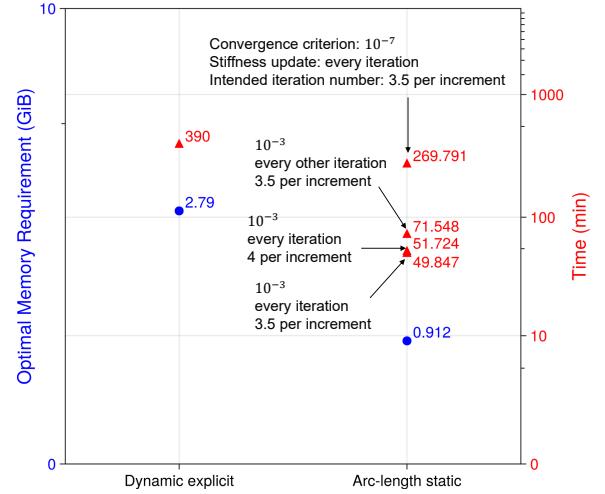
**Figure 14** Validation with three-point bending tests: (a) geometry of notched beams; (b) geometry of unnotched beams; (c) Force-CMOD response for notched beams; (d) Force-pseudo CMOD response for unnotched beams; (e) typical failure mode of notched specimens; and (f) typical failure mode of unnotched specimens. (Crack openings units: mm)

aluminum plates glued on the two sides of the notch. For the unnotched beam, a pseudo CMOD is used, by gluing the plates at the bottom of the beam at a distance of 50 mm on either side from the mid-span. The notched tests were used to calibrate the values of  $\sigma_t$  and  $G_t$ , while the unnotched experiments were used for validation. Overall, the agreement between experimental and numerical results is excellent for both geometries, both in terms of force-CMOD curves (see Figs. 14c and d) and crack patterns at failure (see Figs. 14e and f), which show a predominantly vertical plane of cracking, as observed in the experiments. Moreover, a direct comparison is provided with results reported in Pathirage et al. (2023), with the two approaches exhibiting less than 5% deviation in peak load predictions.

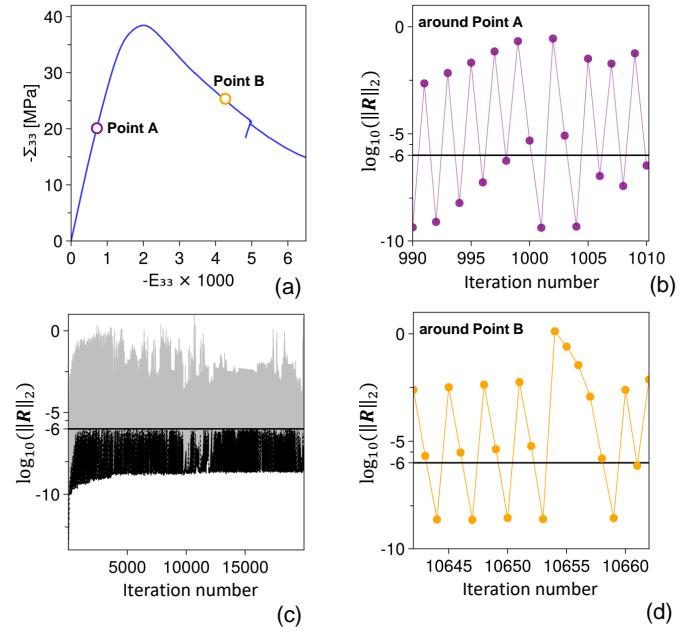
## 6 | QUANTITATIVE ANALYSIS OF MODEL EFFICIENCY

### 6.1 | Computational efficiency

Devising an efficient static solver for LDPM is of great scientific relevance because it allows for analyzing quasi-static fracture crack propagation while enforcing static equilibrium

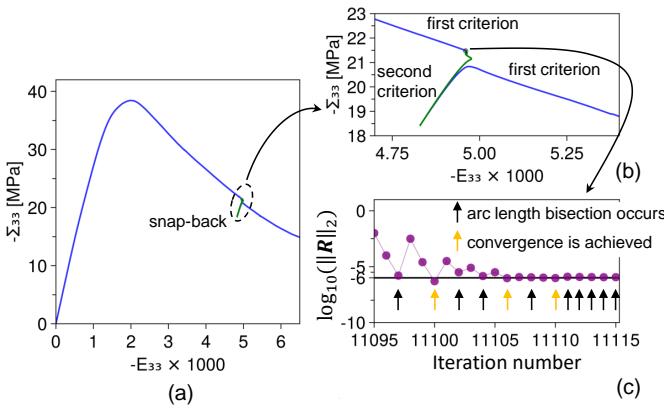


**Figure 15** Comparison of computational efficiency for a three-point bending simulation.

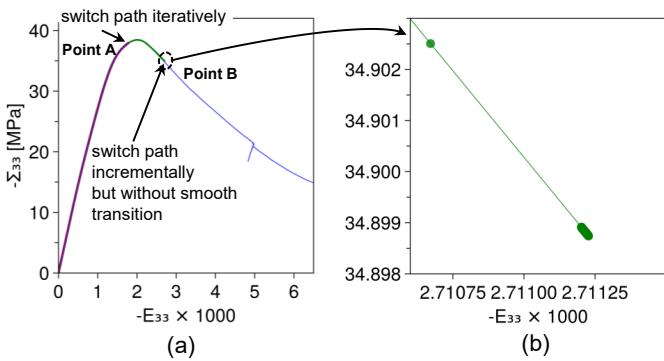


**Figure 16** Convergence analysis: (a) macroscopic stress-strain response for uniaxial compression; (b) convergence at Point A; (c) convergence analysis in the entire loading history; (d) convergence at Point B.

at each step, and at the same time provides fundamental understanding for comparing the obtained solution with the dynamic explicit solvers that have been traditionally employed in the literature (Pathirage, Thierry, et al. 2022). At the same time, adopting a static solver can also allow for significant reductions in computational costs, which traditionally hampered the use of LDPM for large-scale structures. For this reason, this



**Figure 17** Adaptive evaluation of the sign of the iterative load factor: (a) non-convergence on the post-peak branch; (b) switch between criterion #1 (Eq. 35) and #2 (Eq. 36); (c) variation of force residual and bisection of the arc length.



**Figure 18** Example of path switch and transition curve: (a) macroscopic stress-strain response for uniaxial compression; (b) non-convergence attained at Point B when the smooth transition is not enforced at the lattice level.

section reports a quantitative analysis of the computational efficiency of the proposed static solver, by comparison with results previously obtained by means of explicit dynamic solvers.

Pathirage, Thierry, et al. (2022) reported an explicit solution for the three-point bending tests discussed in the previous (see Figs. 14a). The authors investigated both the memory allocation and total computational time required to reach a displacement of 0.15 mm at the loaded point. The dynamic explicit solution used a time step  $\Delta t_{cr} = 2.32 \times 10^{-7}$  s, in accordance with the requirement of time integration stability. Figure 15 reports the comparison between the static solution obtained herein and the explicit dynamic solution reported in Pathirage, Thierry, et al. (2022). All the simulations were performed on a workstation of comparable performance (i.e., Intel(R) Xeon(R) W-2245 CPU @ 3.90 GHz). In the work of Pathirage, Thierry, et al. (2022), the type of memory requirement is not reported. Conservatively,

the presented comparisons assume the reported data as the optimal memory requirement (i.e., the upper bound for memory usage). For the proposed static solver, the memory allocation required for LU factorization of the global stiffness matrix is taken as the optimal memory requirement, which represents the peak allocation in the algorithmic procedure. When evaluating the total computational time, the magnitude of the load increment at each step, the convergence tolerance, and the frequency of updating the Jacobian matrix play a significant role. Therefore, the resultant computational times using different values of these parameters are compared, to provide a holistic picture of the computational efficiency of the proposed solver. As shown in Fig. 15, the proposed static solver can significantly reduce both the optimal memory requirement and the overall computational cost. Concerning memory, the static solver allows for a reduction of approximately 67% for the case examined (0.91 Gb versus 2.79 Gb). Concerning computational time, as expected, employing a larger convergence tolerance while updating the Jacobian at every iteration leads to the least total time consumption of 49 minutes. In contrast, employing a lower tolerance results in the largest total time consumption of 270 minutes. Both values, however, are significantly lower than the time reported to obtain the explicit dynamic solution for the same problem (390 minutes).

## 6.2 | Convergence analysis

One of the fundamental aspects in devising an efficient static solver for LDPM is the rate of convergence of the iterative solution. Due to the complex interaction of material non-linearity, strain softening and damage localization, and the presence of unloading-reloading path, convergence in LDPM simulations poses significant challenges. Such challenges and corresponding remedies include:

1. Discontinuous constitutive laws can lead to difficulties in achieving convergence. Therefore, the LDPM constitutive laws were modified to achieve continuity throughout all possible strain-stress states;
2. Nonlinear and softening constitutive laws necessitates the use of an arc-length method with adaptive bisection;
3. Unloading-reloading paths lead to the frequent exchange between the stress limit and unloading at the element level. A smooth transition between these two branches is introduced to mitigate variations in the Jacobian operator;
4. The use of a history-dependent maximum effective strain parameter (Eq. 19) can induce instabilities and discontinuities in the stress-strain response, causing potential

non-convergence. The proposed adaptive path switch prevents updating of the history variables during iterations at a given increment, facilitating convergence;

5. Equations 32, 33, and 34 can yield a sign for  $\delta\lambda_{k,1}$  for which Eq. 28 has no real solution; however, a real solution may exist if the opposite sign is chosen. The new collaborative criteria (Eqs. 35 and 36) are independent of the properties of the global Jacobian, focusing on the degrees of freedom interested by the external load.

To demonstrate the effects of the proposed techniques, Fig. 16 reports a detailed analysis of the rate of convergence observed in the iterative solution for a simulation of a  $50 \times 50 \times 50$  mm cube undergoing uniaxial compression. Two specific points on the macroscopic stress-strain curves were selected in the pre-peak and a post-peak stages to illustrate the model behavior. Figures 16b and d report the norm of the residual force  $\mathbf{R}$  across successive iterations, which demonstrate quadratic convergence over the majority of the loading history, given a convergence tolerance of  $1 \times 10^{-6}$ . For the post-peak stage, however, only superlinear convergence is attained. This finding highlights both the potential efficiency of the proposed method and the inherent challenges in obtaining static solutions of LDPM governing equations. Moreover, Fig. 16c reports the values of the norm of  $\mathbf{R}$  for the entire loading history.

The effects of adopting the collaborative criteria for the initial value of iterative load factor as well as adaptive bisection of the arc length are demonstrated in Fig. 17, which reports results from the same uniaxial compression numerical test presented earlier. It can be observed that a point of non-convergence is reached in the post-peak response when using criterion #1 (Figs. 17b and c). However, a snap-back instability is followed by criterion #2, as described in the previous. As a reminder, whenever a non-convergent point is detected, bisection of the arc length is performed, as shown in Fig. 17c. When convergence is unattainable after a user-defined number of bisection steps, the adaptive arc-length method enforces a switch in the criterion for determining  $\delta\lambda_{k,1}$ . After this step, the solution is re-attempted on the main softening branch, as demonstrated in Figs. 17a and b.

The adaptive path switch at the lattice element level is key to maintaining a stable system (e.g., stiffness matrix) during iterations within each increment. This method ensures that severe changes in the system only occur between converged increments, significantly aiding convergence of the iterative process. The effects of the adaptive path switch are demonstrated in Fig. 18. Specifically, non-convergence is observed prior to the peak load if the proposed path switch is not enforced (Point A). If the incremental path switch is employed without the proposed smooth transition, a relatively small equilibrium branch

can be reached (Point B). Finally, by employing the proposed smooth transition, the entire stress-strain curve can be obtained.

## 7 | CONCLUSIONS

This manuscript proposed an efficient static solver for LDPM. The solution algorithm incorporates a novel arc-length method that adaptively selects the sign of the initial increment for the load factor, incrementally switches unloading-reloading and stress limit portions of the constitutive law at the lattice level, and employs coloring-aided automatic differentiation to compute the Jacobian. Through the simulation of experiments with varying loading conditions, the proposed static solver exhibits excellent ability to describe complex equilibrium paths with significantly less computational demand than dynamic explicit solutions previously reported for LDPM. The salient findings are summarized as follows:

- (1) The proposed collaborative criterion (Eqs. (35) and (36)) to determine the sign of the initial increments of the load factor alleviates convergence issues in the solution of the LDPM governing equations.
- (2) Restricting path switch between the unloading-reloading and the stress limit paths at converged points is an effective way to mitigate material instabilities.
- (3) A smooth transition between the unloading-reloading path and the stress limit path of elements is required to obtain convergence for macroscopic stress-strain softening branches.
- (4) The use of automatic differentiation and graph coloring provides significant improvements in computational efficiency when evaluating the system Jacobian matrix.
- (5) The proposed implementation achieves 67% reduction in memory requirement and 87% reduction in total computational time when compared to previously reported dynamic explicit solution obtained on computers with comparable specs.

All the analytical and numerical advancements proposed herein can be extended to dynamic quasi-brittle fracture (Adeli 1982; Adeli, Herrmann, & Freund 1977) in future work, bringing the potential to unlock significant computational speed-ups in LDPM simulations.

## References

- Abdellatef, M., Alnaggar, M., Boumakis, G., Cusatis, G., Di Luzio, G., & Wendner, R. (2015). Lattice discrete particle modeling for coupled concrete creep and shrinkage using the solidification microprestress theory. In *Concreep 10* (pp. 184–193).
- Adeli, H. (1982). Dynamic fracture of beam under transverse load. *Journal of the Engineering Mechanics Division*, 108(6), 1025–1035.

- 1131 Adeli, H., Herrmann, G., & Freund, L. (1977). Effect of axial  
1132 force on dynamic fracture of a beam or plate in pure  
1133 bending. *Journal of Applied Mechanics*, 44(4), 647–651.  
1134
- 1135 Alnaggar, M., Cusatis, G., & Di Luzio, G. (2013). Lattice  
1136 discrete particle modeling (ldpm) of alkali silica reaction  
1137 (asr) deterioration of concrete structures. *Cement and  
Concrete Composites*, 41, 45–59.  
1138
- 1139 Alnaggar, M., Pelessone, D., & Cusatis, G. (2019). Lattice  
1140 discrete particle modeling of reinforced concrete flexural  
1141 behavior. *Journal of Structural Engineering*, 145(1),  
04018231.  
1142
- 1143 Angiolilli, M., Gregori, A., Pathirage, M., & Cusatis, G. (2020).  
1144 Fiber reinforced cementitious matrix (frcm) for strength-  
1145 ening historical stone masonry structures: Experiments  
and computations. *Engineering Structures*, 224, 111102.  
1146
- 1147 Ashari, S. E., Buscarnera, G., & Cusatis, G. (2017). A lattice  
1148 discrete particle model for pressure-dependent inelas-  
1149 ticity in granular rocks. *International Journal of Rock  
Mechanics and Mining Sciences*, 91, 49–58.  
1150
- 1151 Bazant, Z. P., Kazemi, M. T., Hasegawa, T., & Mazars, J. (1991).  
1152 Size effect in brazilian split-cylinder tests. measurements  
1153 and fracture analysis. *ACI Materials Journal*, 88(3),  
325–332.  
1154
- 1155 Bažant, Z. P., & Prat, P. C. (1988). Microplane model for  
1156 brittle-plastic material: I. theory. *Journal of Engineering  
Mechanics*, 114(10), 1672–1688.  
1157
- 1158 Bellini, P., & Chulya, A. (1987). An improved automatic  
1159 incremental algorithm for the efficient solution of non-  
1160 linear finite element equations. *Computers & Structures*,  
26(1-2), 99–110.  
1161
- 1162 Belytschko, T., & Hughes, T. J. (1983). *computational methods  
for transient analysis*. Elsevier Science Publishers B.V.  
1163
- 1164 Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017).  
1165 Julia: A fresh approach to numerical computing. *SIAM re-  
view*, 59(1), 65–98. Retrieved from <https://doi.org/10.1137/141000671>  
1166
- 1167 Cibelli, A., Ferrara, L., & Di Luzio, G. (2023). Hygro-thermo-  
1168 chemo-mechanical coupled discrete model for the self-  
1169 healing in ultra high performance concrete. In *Life-cycle  
1170 of structures and infrastructure systems* (pp. 1209–1216).  
1171 CRC Press.  
1172
- 1173 Coleman, T. F., & Moré, J. J. (1983). Estimation of sparse jaco-  
1174 bian matrices and graph coloring blems. *SIAM journal  
on Numerical Analysis*, 20(1), 187–209.  
1175
- 1176 Crisfield, M. A. (1981). A fast incremental/iterative solu-  
1177 tion procedure that handles “snap-through”. In *Com-  
1178 putational methods in nonlinear structural and solid  
mechanics* (pp. 55–62). Elsevier.  
1179
- 1180 Cusatis, G., Bažant, Z. P., & Cedolin, L. (2003a). Confinement-  
1181 shear lattice model for concrete damage in tension and  
compression: II. computation and validation. *Journal of  
Engineering Mechanics*, 129(12), 1449–1458.  
1182
- 1183 Cusatis, G., Bažant, Z. P., & Cedolin, L. (2003b). Confinement-  
1184 shear lattice model for concrete damage in tension  
1185 and compression: I. theory. *Journal of engineering  
mechanics*, 129(12), 1439–1448.  
1186
- 1187 Cusatis, G., Di Luzio, G., & Cedolin, L. (2011). Meso-scale  
1188 simulation of concrete: Blast and penetration effects and  
1189 aar degradation. *Applied Mechanics and Materials*, 82,  
75–80.  
1190
- 1191 Cusatis, G., Mencarelli, A., Pelessone, D., & Baylot, J. (2010).  
1192 The lattice discrete particle model (ldpm) for the simu-  
1193 lation of uniaxial and multiaxial behavior of concrete:  
1194 Recent results. *Fracture Mechanics of Concrete and Con-  
crete Structures-Recent Advances in Fracture Mechanics  
of Concrete*, 509–515.  
1195
- 1196 Cusatis, G., Mencarelli, A., Pelessone, D., & Baylot, J. (2011).  
1197 Lattice discrete particle model (ldpm) for failure behavior  
1198 of concrete. ii: Calibration and validation. *Cement and  
1199 Concrete composites*, 33(9), 891–905.  
1200
- 1201 Cusatis, G., Pelessone, D., & Mencarelli, A. (2011). Lattice  
1202 discrete particle model (ldpm) for failure behavior of  
1203 concrete. i: Theory. *Cement and Concrete Composites*,  
33(9), 881–890.  
1204
- 1205 de Souza Neto, E. A., Peric, D., & Owen, D. R. (2011). *Compu-  
tational methods for plasticity: theory and applications*.  
1206 John Wiley & Sons.  
1207
- 1208 Eliáš, J., & Cusatis, G. (2022). Homogenization of discrete  
1209 mesoscale model of concrete for coupled mass transport  
1210 and mechanics by asymptotic expansion. *Journal of the  
Mechanics and Physics of Solids*, 167, 105010.  
1211
- 1212 Eliáš, J., Yin, H., & Cusatis, G. (2022). Homogenization of  
1213 discrete diffusion models by asymptotic expansion. *Inter-  
national Journal for Numerical and Analytical Methods  
1214 in Geomechanics*, 46(16), 3052–3073.  
1215
- 1216 Fascetti, A. (2016). *Random lattice particle modeling of frac-  
ture processes in cementitious materials* (Unpublished  
1217 doctoral dissertation). Sapienza University of Rome.  
1218
- 1219 Fascetti, A., Bolander, J. E., & Nisticò, N. (2018). Lattice dis-  
crete particle modeling of concrete under compressive  
1220 loading: Multiscale experimental approach for parame-  
1221 ter determination. *Journal of Engineering Mechanics*,  
144(8), 04018058.  
1222
- 1223 Fascetti, A., Feo, L., Nisticò, N., & Penna, R. (2016). Web-  
1224 flange behavior of pultruded grfp i-beams: A lattice  
1225 model for the interpretation of experimental results.  
*Composites Part B: Engineering*, 100, 257–269.  
1226
- 1227 Fascetti, A., Ichimaru, S., & Bolander, J. E. (2022). Stochastic  
1228 lattice discrete particle modeling of fracture in pervi-  
ous concrete. *Computer-Aided Civil and Infrastructure  
Engineering*, 37(14), 1788–1808.  
1229
- 1230 Fascetti, A., & Oskay, C. (2019). Dual random lattice modeling  
1231

- of backward erosion piping. *Computers and Geotechnics*, 105, 265–276.
- Gan, Y., Romero Rodriguez, C., Zhang, H., Schlangen, E., van Breugel, K., & Šavija, B. (2021). Modeling of microstructural effects on the creep of hardened cement paste using an experimentally informed lattice model. *Computer-Aided Civil and Infrastructure Engineering*, 36(5), 560–576.
- Gebremedhin, A. H., Manne, F., & Pothen, A. (2005). What color is your jacobian? graph coloring for computing derivatives. *SIAM review*, 47(4), 629–705.
- George, A., Liu, J., & Ng, E. (1981). *Computer solution of sparse linear systems*. Prentice Hall.
- Grégoire, D., Rojas-Solano, L. B., & Pijaudier-Cabot, G. (2013). Failure and size effect for notched and unnotched concrete beams. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(10), 1434–1452.
- Hassard, B. D., Kazarinoff, N. D., & Wan, Y.-H. (1981). *Theory and applications of hopf bifurcation* (Vol. 41). CUP Archive.
- Kupfer, H., Hilsdorf, H. K., & Rusch, H. (1969). Behavior of concrete under biaxial stresses. In *Journal proceedings* (Vol. 66, pp. 656–666).
- Lale, E., Rezakhani, R., Alnaggar, M., & Cusatis, G. (2018). Homogenization coarse graining (hcg) of the lattice discrete particle model (ldpm) for the analysis of reinforced concrete structures. *Engineering Fracture Mechanics*, 197, 259–277.
- Li, W., Zhou, X., Carey, J. W., Frash, L. P., & Cusatis, G. (2018). Multiphysics lattice discrete particle modeling (m-ldpm) for the simulation of shale fracture permeability. *Rock Mechanics and Rock Engineering*, 51, 3963–3981.
- Mercuri, M., Pathirage, M., Gregori, A., & Cusatis, G. (2023). Fracturing and collapse behavior of masonry vaulted structures: a lattice-discrete approach. *Procedia Structural Integrity*, 44, 1276–1283.
- Ni, T., Zaccariotto, M., Zhu, Q.-Z., & Galvanetto, U. (2019). Static solution of crack propagation problems in peridynamics. *Computer Methods in Applied Mechanics and Engineering*, 346, 126–151.
- Pathirage, M., Thierry, F., Tong, D., Cusatis, G., Grégoire, D., & Pijaudier-Cabot, G. (2022). Comparative investigation of dynamic implicit and explicit methods for the lattice discrete particle model. In *Computational modelling of concrete and concrete structures* (pp. 503–509). CRC press.
- Pathirage, M., Tong, D., Thierry, F., Cusatis, G., Grégoire, D., & Pijaudier-Cabot, G. (2022). Numerical modeling of concrete fracturing and size-effect of notched beams. In *Computational modelling of concrete and concrete structures* (pp. 496–502). CRC press.
- Pathirage, M., Tong, D., Thierry, F., Cusatis, G., Grégoire, D., & Pijaudier-Cabot, G. (2023). Discrete modeling of concrete failure and size-effect. *Theoretical and Applied Fracture Mechanics*, 124, 103738.
- Pelessone, D. (2005, September). *Discrete particle method* (Tech. Rep.). San Diego, USA: ES3 Inc.
- Rall, L. B., & Corliss, G. F. (1996). An introduction to automatic differentiation. *Computational Differentiation: Techniques, Applications, and Tools*, 89, 1–18.
- Shen, L., Di Luzio, G., Zhu, D., Yao, X., Cusatis, G., Cao, M., ... Ren, Q. (2021). Mechanical responses of steel fiber-reinforced concrete after exposure to high temperature: Experiments and mesoscale discrete modeling. *Journal of Engineering Mechanics*, 147(11), 04021084.
- Shen, L., Zhang, H., Di Luzio, G., Yin, H., Yang, L., & Cusatis, G. (2022). Mesoscopic discrete modeling of multiaxial load-induced thermal strain of concrete at high temperature. *International Journal of Mechanical Sciences*, 232, 107613.
- Sherzer, G. L., Alghalandis, Y. F., & Peterson, K. (2022). Introducing fracturing through aggregates in ldpm. *Engineering Fracture Mechanics*, 261, 108228.
- Vonk, R. A. (1992). Softening of concrete loading in compression. *Ph. D. Thesis*.
- Wan-Wendner, L., Wan-Wendner, R., & Cusatis, G. (2018). Age-dependent size effect and fracture characteristics of ultra-high performance concrete. *Cement and Concrete Composites*, 85, 67–82.
- Welsh, D. J., & Powell, M. B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1), 85–86.
- Yip, M., Mohle, J., & Bolander, J. (2005). Automated modeling of three-dimensional structural components using irregular lattices. *Computer-Aided Civil and Infrastructure Engineering*, 20(6), 393–407.
- Zhu, Z., Pathirage, M., Wang, W., Troemner, M., & Cusatis, G. (2022). Lattice discrete particle modeling of concrete under cyclic tension–compression with multi-axial confinement. *Construction and Building Materials*, 352, 128985.
- Zhu, Z., Troemner, M., Wang, W., Cusatis, G., & Zhou, Y. (2023). Lattice discrete particle modeling of the cycling behavior of strain-hardening cementitious composites with and without fiber reinforced polymer grid reinforcement. *Composite Structures*, 322, 117346.

