

# Algorithmic aspects of data-driven computational damage mechanics and introduction of regularization

Vasudevan Kamasamudram<sup>a,b,1</sup>, Laurent Stainier<sup>b</sup>

<sup>a</sup>Institute of Mechanics, Materials and Civil Engineering (iMMC), UCLouvain, Louvain-la-Neuve, B-1348, Belgium

<sup>b</sup>Institut de Recherche en Génie Civil et Mécanique (UMR 6183), Nantes Université - École Centrale de Nantes, Nantes, France

---

## ARTICLE INFO

**Keywords:**

P

---

## ABSTRACT

Data-driven computational mechanics enables the direct use of material databases within finite element analysis through an alternate minimization between mechanical and material states. This work investigates the behavior of this algorithm in the presence of damage-induced softening and snapback instabilities. By analytically prescribing the material response, database discretization effects are eliminated and the algorithmic behavior is isolated. The results show that tangent-consistent hyper-parameters and arc-length continuation are essential to achieve convergence and to follow post-peak response. A regularization strategy introducing an intrinsic length scale is then proposed and its effectiveness and implications are demonstrated through one- and two-dimensional numerical examples.

---

## 1. Introduction

The paradigm of Data-Driven Computational Mechanics (DDCM) was introduced in [19] as an alternative to classical constitutive modeling, replacing explicit material laws with a database of stress-strain pairs obtained, for instance, from experiments or multiscale simulations. In this framework, the solution of the boundary value problem is obtained by minimizing a distance functional defined between admissible mechanical states and material data. While the original work focused on linear elasticity, the approach has since been extended to inelastic problems, including plasticity and viscoelasticity [10], as well as to non-standard continua [18]. For higher-dimensional problems and for history-dependent materials, the application of DDCM requires specialized data structures and sampling strategies, which have been investigated in [9, 31].

Regularization plays a crucial role in DDCM, particularly when localization phenomena occur. In [18], DDCM was employed to predict the onset of shear bands in granular materials, and mesh-size dependence was mitigated by modeling the material as a Cosserat medium. This approach introduces additional kinematic and kinetic variables, together with a corresponding database for higher-order stresses and strains. While effective for shear-dominated problems, such regularization strategies are known to be inadequate for situations where shear is not the dominant deformation mode, such as mode-I crack propagation. This limitation motivates the exploration of alternative regularization strategies within the DDCM framework, especially for fracture and damage problems.

The application of DDCM to fracture has been explored in several studies. In [6], fracture was modeled using a database relating energy release rate to crack length, with the crack evolution predicted through a data-driven minimization procedure; this approach was later extended to rate-dependent fracture in [5]. More recently, [33] combined DDCM with damage mechanics, introducing a micromorphic regularization to achieve mesh-independent predictions. In this setting, a length scale is incorporated through strain gradients, leading to the appearance of couple stresses, as commonly encountered in higher-order continuum theories [24, 34]. Consequently, the material database and the distance functional must be extended to include Cauchy stress-couple stress and strain-strain gradient pairs. This increases the dimensionality of the material phase space, which may lead to increased difficulties in sampling the database. The algorithmic behavior of such formulations in the presence of damage-induced instabilities remains largely unexplored.

Damage and fracture problems are often characterized by strong non-convexities and instabilities such as snapback. In conventional (non-data-driven) formulations, solution strategies typically exploit the separate convexity of the

---

\*Corresponding author

 vasudevan.kamasamudram@uclouvain.be (V. Kamasamudram); laurent.stainier@ec-nantes.fr (L. Stainier)

ORCID(s):

energy functional with respect to displacement and damage variables, enabling robust alternating minimization schemes. In such settings, displacement control is often sufficient to trace the equilibrium path, even in the presence of snapback. In contrast, the behavior of DDCM algorithms under damage-induced instabilities is far less understood. In the data-driven setting, the non-convexity of the strain energy is not expressed explicitly but is instead implicitly embedded in the material database. To date, investigations of instabilities in DDCM have largely focused on geometric nonlinearity. For instance, [20] studied a truss system exhibiting snap-through instability and analyzed the sensitivity of the DDCM solution path to perturbations and algorithmic parameters near limit points. However, corresponding studies addressing instabilities arising from material softening and damage are still lacking.

Another important aspect influencing the performance of DDCM algorithms is the choice of hyper-parameters, particularly the metric used in the distance functional. Previous works have shown that an appropriate choice of metric can significantly accelerate convergence. In [29], a locally linear reconstruction of the material database was used to define an effective local material response, while [25] demonstrated, using a truss example, that optimal convergence rates are obtained when the metric is chosen consistently with the tangent stiffness inferred from the database. These studies, however, focus on elastic-plastic or nonlinear elastic problems, for which the DDCM formulation remains well behaved. In such cases, the choice of metric primarily affects the rate of convergence. As will be shown in this work, the situation is fundamentally different in the presence of damage-induced non-convexity: here, the choice of hyper-parameters can critically influence not only the convergence rate but also whether the DDCM algorithm converges at all, and whether the obtained solution is physically meaningful.

Combining these considerations, the objective of the present article can be established as follows. First, it is essential to analyze the alternate minimization procedure employed in the DDCM setting when softening effects are present. This, in turn, requires a suitable description of the material database incorporating softening behavior. Unlike [33], the present study assumes the availability of a database containing only classical stress-strain pairs, for instance obtained from experiments or lower-scale simulations that capture softening phenomena [28, 27, 26]. Regularization will be introduced separately from the material database. The database is assumed to be stored in a form that appropriately accounts for irreversibility effects; the specific data structures required for this purpose are not addressed here and can be found in [11, 9]. At the same time, it is well recognized that the performance of DDCM algorithms depends on the resolution of the material phase space, and that their sensitivity to database size remains largely unexplored in the presence of softening and instability phenomena. To isolate algorithmic effects from database discretization issues, throughout this article, the material response is therefore described implicitly through an underlying constitutive model, while the solution is still obtained by minimizing the data-driven distance functional with respect to the mechanical and the material states.

The article proceeds incrementally and the objectives of this work are:

- to investigate the convergence behavior of the DDCM alternate minimization algorithm in the presence of damage-induced softening and snapback instabilities, without any regularization in a simple setting;
- to analyze the influence of hyper-parameters on convergence and on the correctness of the obtained solution, without any regularization;
- to introduce a regularization strategy within the DDCM framework suitable for damage and fracture problems, first in 1D and then in 2D;
- to assess the proposed approach through a series of numerical examples involving damage and fracture.

The remainder of this article is organized as follows. First, the DDCM algorithm is briefly reviewed. A one-dimensional example is then introduced to analyze convergence properties in the presence of damage-induced non-convexity. Subsequently, the proposed regularization strategy is presented and extended to two-dimensional problems, followed by numerical examples illustrating the performance of the method.

## 2. Data driven solver

In this section, the data driven solver [19] will be introduced briefly. For this purpose, the notions of mechanical and material states are first introduced. A stress-strain state,  $(\sigma, \epsilon)$ , is called a mechanical state if the stress state,  $\sigma$ , satisfies the conditions of equilibrium and the strain state,  $\epsilon$ , satisfies the compatibility conditions, and the displacement fields satisfy the Dirichlet boundary conditions. At each point in the body, the material state,  $(\sigma^*, \epsilon^*)$ , belongs to the material

manifold,  $\mathcal{D}$ , typically obtained from experiments or from simulations performed at lower scales. The equilibrium and compatibility manifold is defined as

$$\mathcal{E} = \{(\sigma, \epsilon) \mid \sigma, \epsilon \text{ are sufficiently smooth, } \operatorname{div} \sigma = 0 \text{ and } \epsilon = \operatorname{sym} \nabla u, u = u_D \text{ on } \partial \Omega_d\}. \quad (1)$$

The solution is sought as the intersection of the two manifolds.

$$\text{Sol} \in \mathcal{D} \cap \mathcal{E}. \quad (2)$$

In case where the material behavior is defined by a discrete collection of stress-strain pairs, a strict intersection may not exist between the two manifolds. In such a case, the solution is sought as the mechanical-material states that are closest to each other with respect to a certain distance. In discrete case (such as in Finite Element Method, where each element in the mesh is associated with a pair of mechanical and material states), the distance is defined as

$$d(\epsilon, \sigma, \epsilon^*, \sigma^*) = \frac{1}{2}(\epsilon - \epsilon^*)^T C \mathbf{w}(\epsilon - \epsilon^*) + \frac{1}{2}(\sigma - \sigma^*)^T C^{-1} \mathbf{w}(\sigma - \sigma^*), \quad (3)$$

where  $\mathbf{w}$  is vector containing the areas of elements, and  $C$  is a hyper-parameter, typically positive definite.  $\epsilon, \sigma, \epsilon^*, \sigma^* \in \mathbb{R}^{3N}$  in 2D,  $N$  denotes the number of elements, and  $\mathbf{w} \in \mathbb{R}^{3N}$ .  $\mathbf{w}\sigma$  thus indicates an element-wise operation. The discrete equilibrium manifold is defined as

$$\mathcal{E} = \{(\sigma, \epsilon) \mid \mathbf{B}^T \mathbf{w}\sigma = \mathbf{0} \text{ and } \epsilon = \mathbf{B}u, u = u_D \text{ on } \partial \Omega_d\}. \quad (4)$$

In the above,  $\mathbf{B} \in \mathbb{R}^{3N \times 2N}$  denotes the discrete strain-displacement operator,  $\mathbf{u} \in \mathbb{R}^{2N}$  is the displacement vector. Thus, the solution is the minimizer of the distance functional.

$$\text{Sol} \in \arg \min d(\epsilon, \sigma, \epsilon^*, \sigma^*) \text{ s.t. } (\sigma, \epsilon) \in \mathcal{E} \text{ and } (\sigma^*, \epsilon^*) \in \mathcal{D}. \quad (5)$$

The minimization can be carried out by imposing the constraints using Lagrange multipliers. The minimization process is divided into mechanical and material updates [19]. During the mechanical update, the material states, denoted by the asterisked quantities are held fixed. The minimization is carried out wrt the mechanical states. During the material update, the mechanical states are held fixed and the minimization is carried out wrt material states, typically element wise. The corresponding Euler Lagrange equations during the mechanical update can be seen to be:

$$\mathbf{B}^T C \mathbf{w} \mathbf{B} \mathbf{u} = \mathbf{B}^T C \mathbf{w} \epsilon^*, \quad (6)$$

$$\sigma - \sigma^* = C \mathbf{B} \eta, \quad (7)$$

$$\mathbf{B}^T \mathbf{w} \sigma = \mathbf{0}. \quad (8)$$

The last two of the above equations are combined and written as

$$\mathbf{B}^T \mathbf{w} C \mathbf{B} \eta = \mathbf{B}^T \mathbf{w} \sigma^*. \quad (9)$$

$\eta$  is treated as the perturbation on the displacement field and so is taken as  $\mathbf{0}$  on the Dirichlet boundary. Equation 9 is used to obtain the remaining degrees of freedom.

The material update in the DD framework is performed at each Gauss point by searching for the point in the material database that is nearest to the current mechanical state. This step can be significantly optimized by employing specialized data structures for database storage and search, as discussed in [12]. The mechanical and the material updates are performed repeatedly until convergence. Refinement of the database ultimately leads to the convergence of the DD solution to the corresponding model-based solution [8].

The present study focuses on extending the DDCM paradigm to problems involving progressive damage and fracture. Such problems are characterized by strong non-convexities arising from material softening. In conventional, model-based formulations, these non-convexities are typically handled through alternating minimization schemes that exploit the separate convexity of the energy functional with respect to displacement and damage fields, even in the presence of snapback. In contrast, the DD approach bypasses explicit constitutive modeling and embeds material

nonlinearity implicitly through the material database. While it is known that DD solutions converge to their model-based counterparts in the limit of an infinitely rich database, the behavior of DD algorithms in the presence of damage-induced non-convexity and softening remains largely unexplored.

To investigate these effects in a controlled setting where the material phase space discretization effects are removed, the present work adopts a constitutive model-informed data-driven framework. **Instead of relying on a finite and potentially poorly sampled database, the material update is performed using a known constitutive relation, which acts as an implicit, infinitely rich database.** At each material point, the stress-strain pair is obtained through an analytical or numerical minimization of the data-driven distance functional subject to the constitutive relation. This approach is equivalent to a DD computation with an infinite dataset and eliminates errors associated with insufficient sampling of the material phase space. As a result, it allows the fundamental algorithmic challenges of applying DD mechanics based alternate minimization strategy to damage and fracture, particularly those related to non-convexity, softening, and convergence, to be isolated and examined.

Subsequently, a 1D problem without any regularization is considered first. The underlying constitutive model for the material database will allow a simple analysis of the 1D problem in various cases. Following this, the demonstration will be extended to the 2D case, with a regularization technique introduced later on.

### 3. Algorithmic robustness and convergence analysis

In this section, the DD algorithm described above is first applied to a one-dimensional problem to investigate the behavior of a material undergoing damage. In conventional simulations based on constitutive models without an intrinsic length scale, damage is known to localize within a single element, leading to pronounced mesh dependence as the localization width decreases with mesh refinement. To obtain mesh-objective results, such models typically introduce a length scale, for instance through gradients of strain or damage. For consistency, the DD algorithm should exhibit similar qualitative behavior: in the absence of an intrinsic length scale, its predictions should likewise depend on the mesh size, localizing in a single element, under appropriate loading and initial conditions.

However, since the DD solution is obtained through the minimization of a distance functional, with non-convexity embedded directly in the material database, it is not a priori clear whether damage will localize within a single element. This possibility is examined in the following discussion using a data-driven formulation in which the material behavior is provided implicitly by a local damage model. As noted at the end of the previous section, the material update is performed directly using the analytical expression of the constitutive model, as detailed in Appendix C. Briefly, at each Gauss point, the material update is performed as

$$(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) = \arg \min_{(\tilde{\boldsymbol{\epsilon}}^*, \tilde{\boldsymbol{\sigma}}^*)} d(\boldsymbol{\epsilon}, \boldsymbol{\sigma}, \tilde{\boldsymbol{\epsilon}}^*, \tilde{\boldsymbol{\sigma}}^*) \quad \text{s.t.} \quad \tilde{\boldsymbol{\sigma}}^* = f(\tilde{\boldsymbol{\epsilon}}^*), \quad (10)$$

where  $f$  is the constitutive model that relates the material stresses and strains. Then,

$$\frac{\partial}{\partial \tilde{\boldsymbol{\epsilon}}^*} \tilde{d}(\tilde{\boldsymbol{\epsilon}}^*, f(\tilde{\boldsymbol{\epsilon}}^*); \boldsymbol{\epsilon}, \boldsymbol{\sigma}) = 0, \quad (11)$$

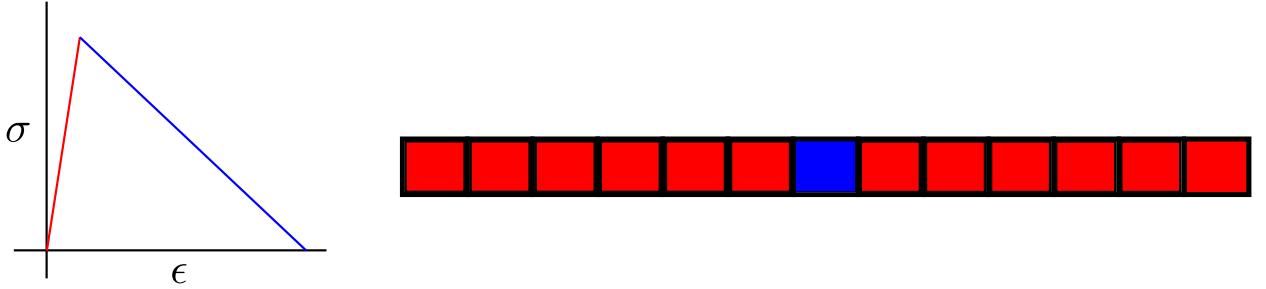
where  $\tilde{d}(\tilde{\boldsymbol{\epsilon}}^*, f(\tilde{\boldsymbol{\epsilon}}^*); \boldsymbol{\epsilon}, \boldsymbol{\sigma}) = d(\boldsymbol{\epsilon}, \boldsymbol{\sigma}, \tilde{\boldsymbol{\epsilon}}^*, \tilde{\boldsymbol{\sigma}}^*)$ . The mechanical update is left as is. The overall algorithm has been presented in 1, explained later.

The constitutive model, serving here as an implicit database, is represented by a set of piecewise affine relations:  $\boldsymbol{\sigma} = E\boldsymbol{\epsilon}$  for the elastic regime,  $\boldsymbol{\sigma} = m\boldsymbol{\epsilon} + \boldsymbol{c}$  ( $m < 0$ ) for the softening regime, and  $\boldsymbol{\sigma} = E_i\boldsymbol{\epsilon}$  for the  $i$ th unloading branch (see Figure 9). The mechanical and material updates are presented in A.

With this setting, the following analysis demonstrates that the convergence and correctness of the DD alternate minimization algorithm in the presence of damage-induced non-convexity hinge on two essential ingredients: the choice of hyper-parameters employed during the updates, and the use of a continuation strategy capable of traversing snapback. These conclusions are established using a one-dimensional bar problem, which permits a complete analytical characterization of the associated fixed-point operator and isolates algorithmic effects from geometric or sampling artifacts.

#### 3.1. Convergence of the method and its rate

Since a rigorous convergence proof of the DD algorithm in the presence of damage-induced non-convexity is currently unavailable, an algorithmic analysis will be performed based on representative one-dimensional configurations.



**Figure 1:** 1D bar with damaged element

The study focuses on identifying the mechanisms governing convergence, loss of convergence, and branch selection under softening.

Consider the case where the element in the middle on the damage branch and the rest of the elements unload on the elastic branch. See Figure 1. It is the intent to check the effect of the hyper-parameters in data driven case on the (rate of) convergence of the solution in this case. If the material behavior is described by a local damage model (or sometimes even with a non-local model), a snapback is expected in the load displacement response of the body in some scenarios, depending on the mesh size, and the regularization length, when present. Two cases can thus be considered : one with a snapback and one without. In either case, the stress state is homogeneous in the entire bar as a consequence of equilibrium. The strain is localized in the element undergoing damage (in blue in Figure 1), while the other elements (in red) are unloading. Thus, the mechanical state of the bar is described by a scalar stress  $\sigma \in \mathbb{R}$  and strains  $\epsilon_1, \epsilon_2 \in \mathbb{R}$  ( $\epsilon_1$  for the undamaged elements and  $\epsilon_2$  for the damaged element). The corresponding material states are defined as  $(\epsilon_1^*, \sigma_1^*) \in \mathbb{R}^{1 \times 1}$  for the elements in red and  $(\epsilon_2^*, \sigma_2^*) \in \mathbb{R}^{1 \times 1}$  for the element in blue. The displacement at the right end of the bar is given as  $u_D$  and the left end is fixed. The distance function in this is given by

$$d(\sigma, \sigma_1^*, \sigma_2^*, \epsilon_1, \epsilon_2; u_D) = \frac{N\ell_e}{2} \left[ \frac{(\sigma - \sigma_1^*)^2}{C_1} + C_1(\epsilon_1 - \epsilon_1^*)^2 \right] + \frac{\ell_e}{2} \left[ \frac{(\sigma - \sigma_2^*)^2}{C_2} + C_2(\epsilon_2 - \epsilon_2^*)^2 \right]. \quad (12)$$

The compatibility condition is  $N\epsilon_1 + \epsilon_2 = u_D/\ell_e$ ,  $\ell_e$  is the element length,  $C_1, C_2$  are the hyper-parameters for the undamaged and damaged elements, respectively, and the number of elements is  $N + 1$ . Introducing the compatibility condition into equation 12, Euler-Lagrange equations corresponding to the mechanical update can be obtained by setting the appropriate variations to 0. In the current case, Euler-Lagrange equations can be obtained for material updates as well by setting  $d_{,\epsilon_1^*} = 0$  and  $d_{,\epsilon_2^*} = 0$ .

Defining  $\mathbf{a} = [\epsilon_1, \epsilon_2, \sigma]^T$ ,  $\mathbf{b} = [\epsilon_1^*, \epsilon_2^*, \sigma_1^*, \sigma_2^*]^T$ , as shown in Appendix A, the mechanical and the material updates with in a load increment can be written concisely as

$$\mathbf{b}^{i+1} = \mathbf{Ca}^i + \mathbf{d}, \quad (13)$$

$$\mathbf{a}^{i+1} = \mathbf{Ab}^{i+1} + \mathbf{c}, \quad (14)$$

where  $\mathbf{A}, \mathbf{C}, \mathbf{c}, \mathbf{d}$  are parameters defined in Appendix A.  $i$  denotes the iteration number. The updates can be combined and written together as

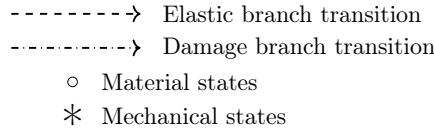
$$\mathbf{a}^{i+1} = \mathbf{ACa}^i + \mathbf{f}. \quad (15)$$

At convergence,  $\mathbf{a}^{i+1} \approx \mathbf{a}^i \approx \mathbf{a}$ , where  $\mathbf{a}$  is the exact solution. Defining the error at step  $i$  as  $\mathbf{e}^i = \mathbf{a}^i - \mathbf{a}$ , the errors at successive iterations can be related, using equation 15, as

$$\mathbf{e}^{i+1} = \mathbf{ACe}^i. \quad (16)$$

Thus the rate of convergence of the method (and whether it converges at all) can be determined by analyzing the matrix  $\mathbf{AC}$ , that corresponds to the combined mechanical and the material updates.

With this framework, the 1D bar in Figure 1 will now be analyzed for various cases . It will be assumed that the solution at some load step  $n$  is known. The external load  $u_D$  is then varied. The intent is to find the solution to the 1D

**Figure 2:** Mechanical and material states, with transitions

problem, given the state at step  $n$ , under the new loading condition. For this purpose, the trajectory of the mechanical and the material states followed by the elements on corresponding branches within the mechanical-material minimization process of the DD algorithm will be presented on a  $\sigma - \epsilon$  space. As represented in Figure 2, the mechanical states are represented by an asterisk (\*) and the material states by a circle (o). Within a DD iteration, the transition between the elements on the elastic branch are indicated by a dashed arrow and that on the softening branch by a dash-dotted arrow.

For all the cases considered below, the material parameters are taken as  $E = 100, m = -7.1428, c = 107.142$ . The initial mechanical and material states are taken as  $\sigma = \sigma_1^* = \sigma_2^* = 99.95, \epsilon_1 = \epsilon_1^* = 0.9995, \epsilon_2 = \epsilon_2^* = 1.007$ . This state corresponds to points that are just below the cusp of the damage initiation point. For the purpose of the analytical convergence study presented in this subsection, elements are temporarily restricted to remain on their respective solution branches (elastic unloading or softening). It is thus expected that the damage level in the blue elements continues to increase while the rest of the elements undergo unloading along the elastic branch. This restriction is introduced solely to isolate the fixed-point properties of the DD alternating minimization near a given bifurcated solution and does not reflect an assumption of the general DD algorithm. In particular, branch restriction is removed in subsequent examples, where the method is allowed to select material states freely.

### **Behavior without snapback under displacement control**

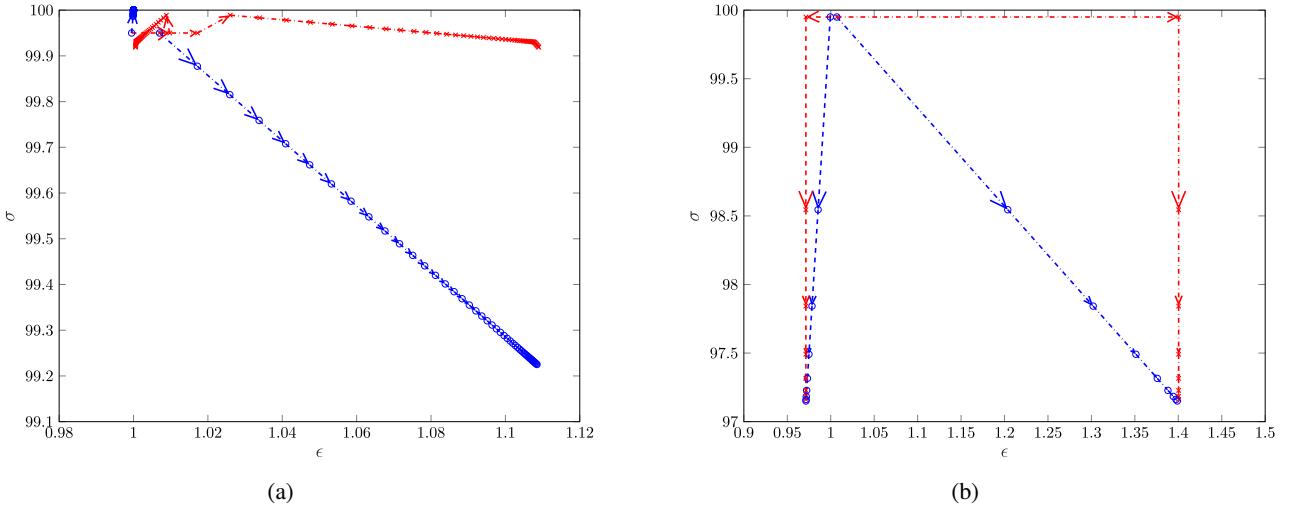
A snapback is not expected to occur when the mesh size, and hence the localization length, is *big enough*. This is the case when  $N = 10$ . Direct displacement control can be used in this case. Thus, the new applied displacement is taken as  $u_D = 1.0104$ . The solution evolution is continuous.

For the first case,  $C_1, C_2$  are taken equal to  $E$ . The successive iterates for this case, computed from equations 14 and 13, are presented in Figure 3a. The zoom-in of the first few iterations is presented in Figure 4. At the first iteration, the mechanical states for all the elements move towards the damage branch. During the subsequent iterations, the mechanical states corresponding to the elements on the elastic branch return back, while the state of the element on the softening branch continues to go further along the branch. With increasing iteration count, the iterates on the elastic branch become progressively closer to one another, and the same behavior is observed for the damage branch. At some point, successive iterates can no longer be distinguished in Figure 3a, even though the mechanical and the material states are still further apart and the distance functional still assumes higher values. The reason behind this observation can be obtained by analyzing the matrix  $\mathbf{AC}$  in equation 15.

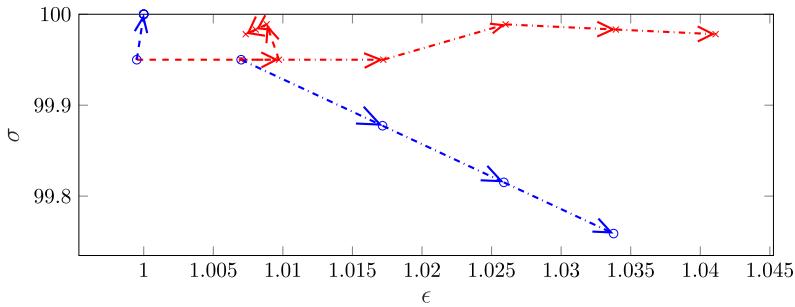
The matrix  $\mathbf{AC}$  is not symmetric. But for the parameters listed above, its eigen values are real and the eigen vectors are linearly independent. Thus, the rate of convergence of the iterations in equation 15 can be obtained from the spectral radius of the matrix  $\mathbf{AC}$ . When  $C_1, C_2$  are taken equal to  $E$  in equations 38 and 39, the spectral radius of  $\mathbf{AC}$  is 0.9994. This implies that the fixed point iterations in equation 15 are contractive and hence converge, albeit at a very slow rate. This explains the closeness of the successive iterates in Figure 3a. If a discrete database were used instead of the analytical expressions, an extremely fine database would be required to ensure that the distance functional reduces sufficiently at convergence.

Following this, the case  $C_2 = m$  is examined to assess how this choice affects the rate of convergence. Since  $m < 0$ , this formally corresponds to assigning a negative weight to the strain term in the distance functional, which would be inadmissible if used to define the metric itself. For this reason,  $C_2 = m$  is employed only in the Euler-Lagrange equations 38 and 39. Convergence can still be evaluated exclusively using the distance functional 12 with  $C_2 = E$ . The choice  $C_2 = m$  therefore does not alter the underlying metric and should be interpreted solely as an algorithmic modification used to probe its effect on convergence behavior.

The matrix  $\mathbf{AC}$  remains unsymmetric with  $C_2 = m$ , but still possesses real eigen values and linearly independent eigen vectors. Its spectral radius in this case is 0.5. The successive iterates of mechanical and material states using equations 13 and 14 are presented in Figure 3b. Unlike the behavior observed in Figure 3a, the iterates are more widely spread. The mechanical and material states approach each other over successive iterations and tend to intersect at a



**Figure 3:** Results from the 1D analysis. Successive mechanical and material states, connected by arrows as indicated in Figure 2. (a)  $C_2 = E$ , no snapback, and without the arc length constraint. (b)  $C_2 = m$ , no snapback, and without the arc length constraint.



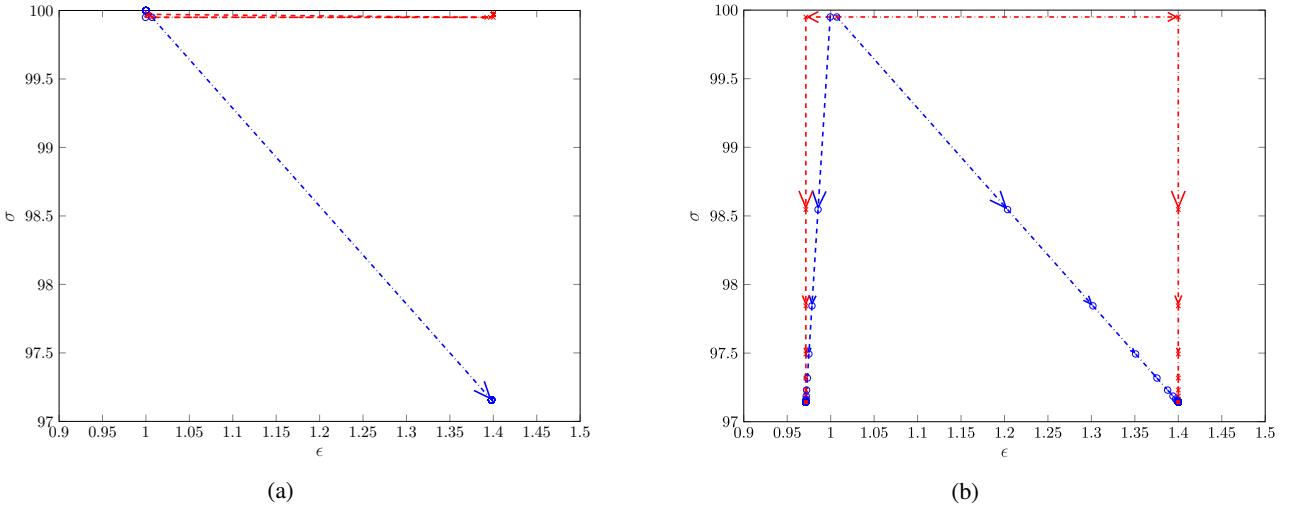
**Figure 4:** Zoom in of the first few iterations from Figure 3a.

rate much faster than when  $C_2 = E$ . The solution converges within just 7 iterations. Also, when  $C_2 = E$ , during the first iteration, the (mechanical) strain changes for the elements on the elastic and the damage branch point in the same direction (positive) as the external load (also positive). In contrast, when  $C_2 = m$ , the sign of the strain change for the element on the damage branch remains the same as the external load, while for the element on the elastic branch, it is opposite.

**Remark.** When  $C_2 = E$ , the initial mechanical strain increment for the elements on the elastic and softening branch are in the same direction. If the elements are not restricted to lie on their corresponding branches during the course of iterations, the system will thus favor the homogeneous branch of the solution, since starting from iteration 2, the nearest points for the material state of the elements in red lie on the damage branch. See Figure 3a. Choosing  $C_2 = m$  will let the system continue on the existing bifurcation branch as presented in Figure 3b. Taking  $C_2 = E$  is thus likely to cause the solution to revert from the bifurcated path to the homogeneous one, an outcome that is undesirable when the objective is to capture localized damage evolution. This will be of considerable importance in 2D, as shown later.

#### Behavior with snapback and arc-length solver

When the element size is decreased, a snapback response is expected to occur. In conventional simulations performed using constitutive models, the alternative minimization performed with respect to the displacement and the damage variables is capable of capturing the snapback, albeit with taking a large number of iterations to converge. The solution typically 'jumps through' the snapback under displacement control. This scenario will be examined when



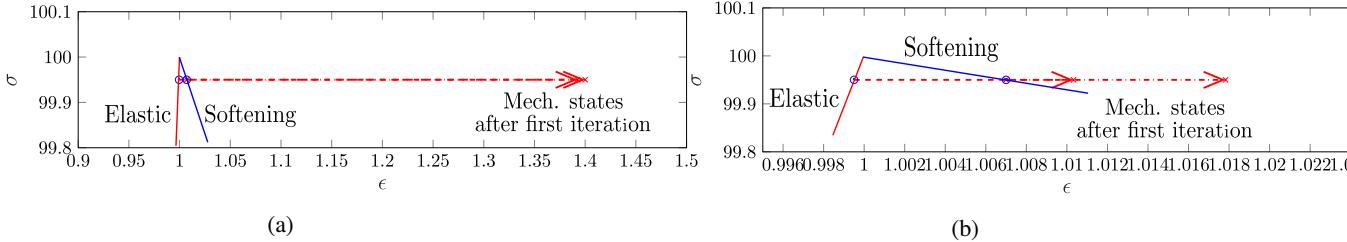
**Figure 5:** Results from the 1D analysis. Successive mechanical and material states, connected by arrows as indicated in Figure 2. (a)  $C_2 = E$ , snapback, and with the arc length constraint. (b)  $C_2 = m$ , snapback, and with the arc length constraint.

using DD algorithm using the example described above, using an arc length solver. The material parameters and the initial conditions are kept the same as in the previous case, but with  $N = 100$ . The solution update using an arc length solver is presented in equations 42 and 41. The arc length constraint corresponds to prescribing the strain value for the element on the damage branch, in the current case as 1.4. Thus, the prescribed displacement  $u_D$  is obtained as a part of the solution.

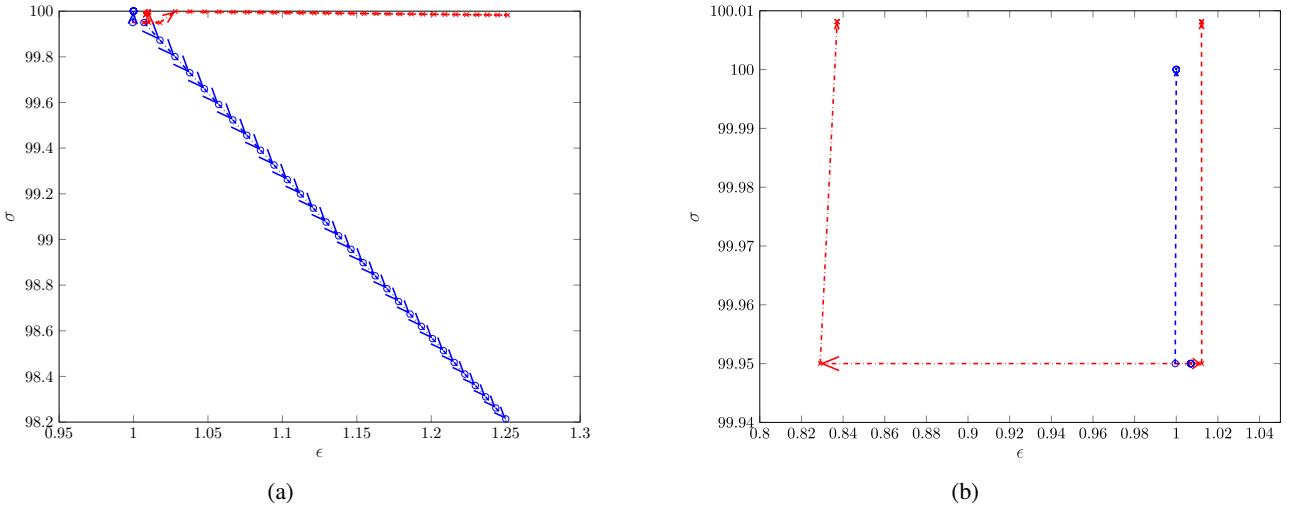
When  $C_2 = E$ , the path taken by the successive iterates are presented in Figure 5a. The arc length constraint carries all the mechanical strains near the prescribed arc-length value of 1.4 in the first iteration. Since the elements on the elastic and the damage branches are forced to remain on their corresponding branches, during the subsequent iterations, the element on the damage branch remains where it was after the first iteration, while the states of the elements restricted to the elastic branch return back. If the branch restriction is removed, the solution migrates to a homogeneous branch, even when the initial iterate (and thus the converged solution from the previous load step) lies slightly away from the elastic-damage cusp. This occurs because, from the second iteration onward, the nearest admissible material state for the elements initially marked in red is located on the damage branch. See Figure 6a, that shows the position of the mechanical states after the first mechanical update. This causes all elements to collapse onto the same branch like in the previous case in Figure 3a. The spectral radius of the matrix  $\mathbf{AC}$  is 1.029, which indicates the unconvvergent nature of the iterations with branch restrictions.

When  $C_2 = m$ , the path taken by the successive iterates are presented in Figure 5b. The strain increments for the elements on the elastic and the softening branches point in the opposite directions, differently from when  $C_2 = E$ . The mechanical and the material states quickly reach the intersection point. In this case, the spectral radius of  $\mathbf{AC}$  is 0.5. This stresses the importance of the hyper-parameters, specifically in case where a snapback is expected to occur. In this case, it is necessary to take  $C_2 = m$  to arrive at a converged solution.

As presented in Figure 5a, if  $C_2 = E$ , the mechanical strain change for both the elements point in the same direction. In subsequent iterations, the same material state is selected for the element on the elastic branch (as it has restricted to remain on the elastic branch in the current case to study the bifurcated solution). As a result, the distance functional stagnates at a high value and does not change anymore regardless of the number of iterations. However, when  $C_2 = m$ , not only is the rate of convergence optimal, but also the algorithm converges, and to the correct solution. This stresses the importance of using an arc-length solver when expecting snapback instabilities in the DDCM paradigm. Not using the proper hyper-parameter - arc-length combination may result in an artificial widening of the damage zone even without any regularization, as a consequence of the trajectories followed by the mechanical and material states during the mechanical and the material updates.



**Figure 6:** Results from the 1D analysis. Successive mechanical and material states, connected by arrows as indicated in Figure 2. (a) Zoom-in of the first iteration in Figure 5a. (b) Zoom-in of the first iteration in Figure 7a.



**Figure 7:** Results from the 1D analysis. Successive mechanical and material states, connected by arrows as indicated in Figure 2. (a)  $C_2 = E$ , snapback, and with the arc length constraint, different starting point. (b)  $C_2 = m$ , snapback, and with the arc length constraint, different starting point.

### Failure with snapback and displacement control

To assess whether an arc-length solver is strictly necessary i.e., whether the data-driven algorithm can intrinsically jump through snapback instabilities, the snapback case is next examined under pure displacement control, without arc-length continuation. The corresponding DD iterations are shown in Figures 7a and 7b.

When  $C_2 = E$ , the mechanical and material states initially progressively drift apart and then stagnate. Thus, the distance functional increases initially and does not change any further later on. For this case, the spectral radius of  $\mathbf{AC}$  is 1. Thus, the component of the error that corresponds to the eigen vector associated with this eigen value does not reduce anymore even after the other components are annihilated. If the branch restriction is relaxed, the algorithm instead collapses rapidly to the homogeneous damage solution: already from the second iteration onward, the closest admissible material state for both elastic and softening elements lies on the damage branch, as illustrated in Figure 6b.

When  $C_2 = m$ , the mechanical and material states change during the first iteration but subsequently stagnate. In this case, the strain in the softening element decreases while that in the elastic elements increases, despite an increasing applied displacement. This behavior is consistent with the loss of positive definiteness of the tangent operator at snapback, reflected by the emergence of a negative eigenvalue. Consequently, the strain increment in the softening element necessarily opposes the imposed displacement increment, as discussed in classical continuation analyses [4, 3, 1]. With branch restriction enforced, this sign inconsistency leads to stagnation of the DD iterations (Figure 7b). If the restriction is removed, the algorithm again transitions to the homogeneous damage solution. In any case, displacement control proves insufficient to traverse snapback within the DD framework, irrespective of the chosen metric, thereby demonstrating the necessity of arc-length continuation for robust tracking of damage-induced instabilities.

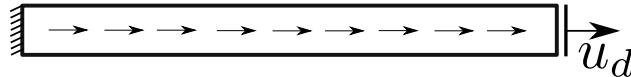


Figure 8: 1D bar with a body force

Thus, it can be briefly concluded that not only is using an optimal value for  $C_2$ , together with an arc length solver, sufficient to achieve an optimal rate of convergence, but it is also necessary to achieve the expected localization in some cases with instabilities. This is in contrast to the study in [25], where the behavior of DD solver is studied for the case of a non-linearly elastic material, an optimal value for  $C_2$  is only used to accelerate the convergence. Also, the paths taken by the iterates in Figures 7a and 7b resemble the successive mechanical iterates taken by a truss system, presented in Figures 6, 7, and 16 of [20]. In that reference, the performance of DD solver has been analyzed for cases involving instabilities. A truss system was considered, with a linearly elastic material behavior. The authors of [20] extract the eigenvector corresponding to the zero eigenvalue of the hessian matrix to perturb the initial condition in DD solution. They also reverse their loading when the hessian matrix develops a zero eigenvalue. This is thus similar to implicitly using an arc-length solver. In the current case, the origins of the instability are both material (softening) and geometric (mesh size/ characteristic length). Thus, an arc-length solver will be used to trace any and all the snapback instabilities.

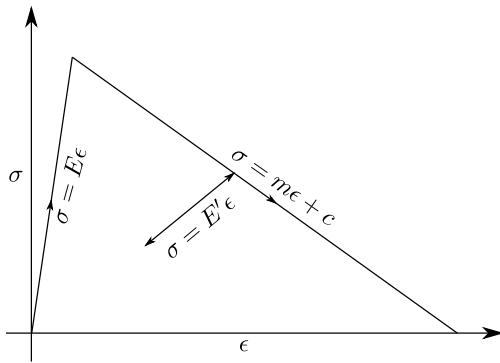
In the present work, the underlying material model is characterized by linear softening, which allows the convergence behavior to be analyzed through the spectral radius of the matrix  $\mathbf{AC}$ . If this condition is not satisfied, for example, if the softening response is nonlinear or the problem is formulated in two dimensions, such an analysis is no longer feasible, since  $\mathbf{AC}$  then depends on the evolving strain state. Nevertheless, the two-dimensional analyses presented later demonstrate that, for the cases considered, the convergence rate is quadratic i.e., the distance between successive iterates is halved at each iteration when the material tangent is used as the hyper-parameter similar to all the convergent 1D cases presented earlier.

### 3.2. A 1D bar with a body force

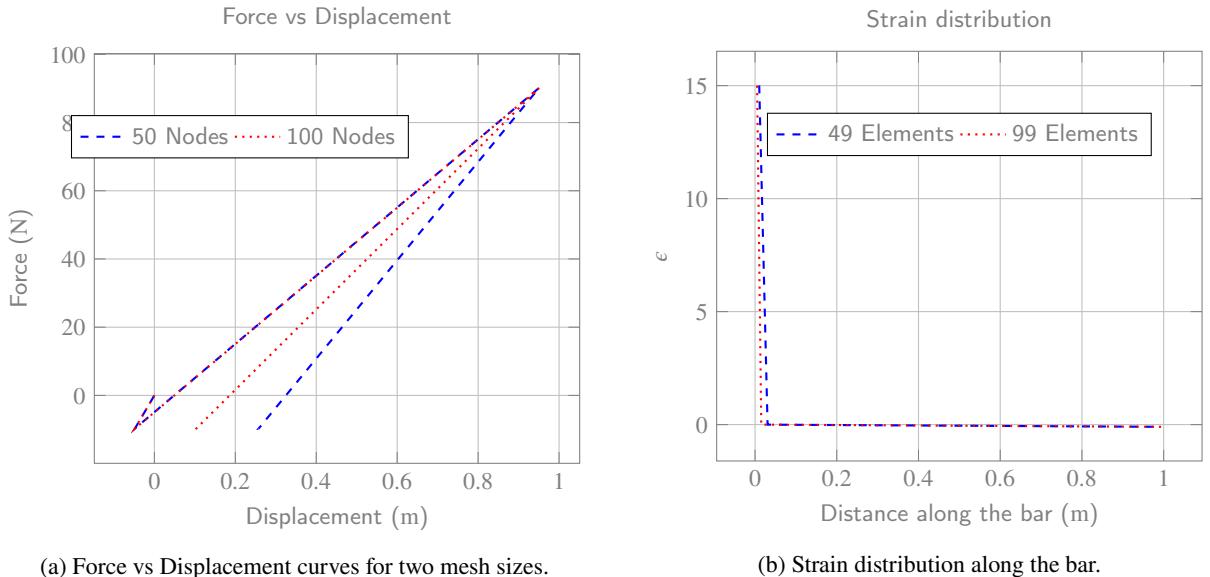
In contrast to the analytical study of Section 3.1, no branch restriction is imposed in the following example. The material update is allowed to select the material state at each iteration as described below, and the observed behavior therefore reflects the intrinsic branch-selection properties of the DD algorithm.

The problem of 1D bar subjected to a body force as in Figure 8 will now be analyzed. The same problem will then be used to demonstrate the effect of introducing a length scale into the problem. The bar is meshed with piecewise affine and continuous displacement elements. The set of equations that govern the data driven solution can be found in the previous section, in equations 6 and 9. The arc-length solver - hyper-parameter combination that was presented earlier will be used to capture the (possible) snapback response of the bar. The hyper-parameter in this case will be the value of the tangent associated to the corresponding material state of an element. Expressing the displacement vector as  $\mathbf{u} = \mathbf{C}_f \mathbf{u}_f + \lambda_g \mathbf{u}_g$ , where  $\mathbf{u}_f$  are the unknown degrees of freedom,  $\lambda_g$  is the strength of the applied boundary condition,  $\mathbf{u}_g$  is a vector that corresponds to the Dirichlet boundary conditions, and  $\mathbf{C}_f$  is a matrix of zeros and ones that relates  $\mathbf{u}_f$  to  $\mathbf{u}$ . Similarly,  $\boldsymbol{\eta} = \mathbf{C}_f \boldsymbol{\eta}_f$ . If the body is loaded through an applied displacement,  $\lambda_g$  is a known and the equations 6 and 9 are solved for  $\mathbf{u}_f$  and  $\boldsymbol{\eta}_f$ . When an arc length solver is used,  $\lambda_g$  is treated as an unknown. The system of equations is supplemented with an additional arc length constraint equation. More details on this can be found in D. The mechanical stresses are taken to be piecewise constant. The mesh size is homogeneous. The problem is examined for two mesh sizes (say  $\ell_e$ ,  $0.5\ell_e$ ) to determine the effect of the mesh size on the response of the bar. The database is taken as in Figure 9 and the corresponding material parameters are same as the ones used to perform the convergence analysis earlier. Material state updates are performed using equations 46 and 47. The update algorithm has been presented in alg 1. The entries of the matrix  $\mathbf{C}$  in equations 6 and 9 are updated depending on the solution branch on which the material state of the element corresponds to, as stated in alg 1.

The force at the right end of the bar vs the applied displacement until complete failure (where some element in the bar is fully damaged) for the two mesh sizes has been presented in Figure 10a. The mesh dependence of the strain distribution, together with the snapback response, can be clearly seen once the *damage begins to increase*. At the first step, a compressive force is developed as a consequence of the body force, which gradually becomes positive as the displacement is increased. The initial negative displacement is a consequence of the arc-length constraint that the strain increment in the left most element is equal to a certain (user specified) value. As the load is increased further,

**Figure 9:** Stress strain database

the damage initiates at the left end (just at the left end because of the applied body force). The strain and the damage localize in the element at the left end in the subsequent steps until the element undergoes a complete failure ( $d = 1$ ). A small compressive load can again be seen at the end as a consequence of the applied body force. The corresponding strain distributions, at complete failure, for the two mesh sizes can be seen in figure 10b. The mesh dependence of the solution is obvious. Using appropriate values for the hyper-parameters, together with the arc-length solver accurately captures the snapback response of the bar, mesh dependence of the solution, and follows the expected bifurcation branch.

**Figure 10:** Results without regularization

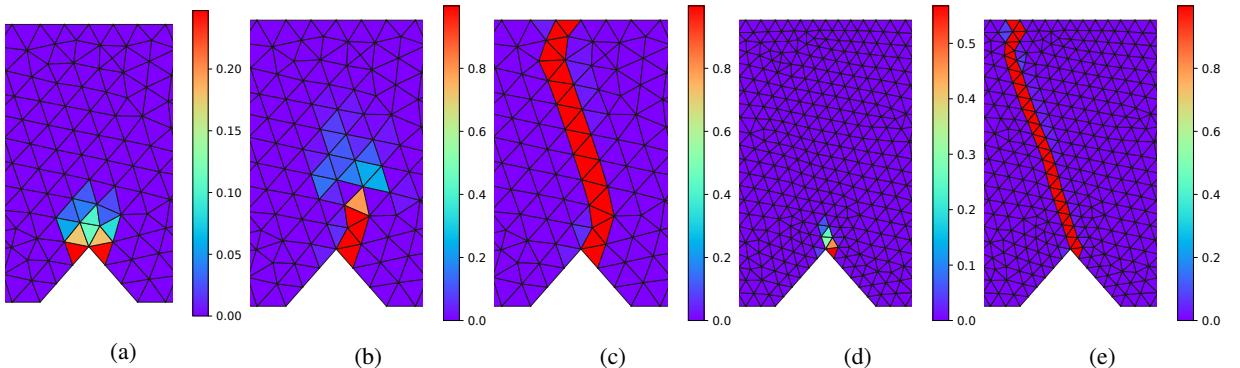
### 3.3. 2D case - Bar with a notch

The tests performed above in 1D case has been repeated in 2D. The example used is same as the one used later on to demonstrate the effect of introducing a length into the problem. The details about the geometry, loading, and the material properties have been presented in section 6.1. The simulations have been performed without introducing any length scale. Thus, the damage is expected to localize over a one element wide region. As earlier, three cases have been analyzed

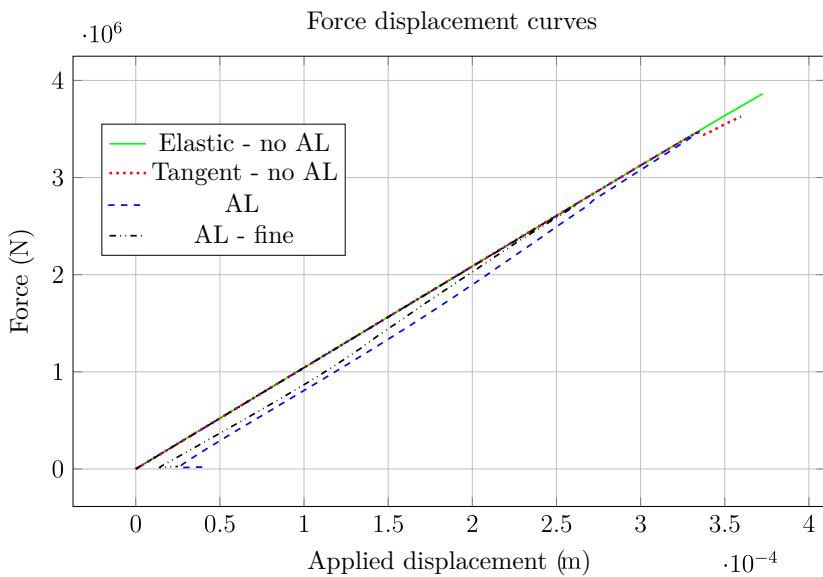
- Degraded elastic moduli of the corresponding material states are taken as the hyper-parameters. The system is displacement driven and an arc-length solver has not been used.

- Tangent moduli of the corresponding material states are taken as the hyper-parameters. The system is displacement driven and an arc-length solver has not been used.
- Tangent moduli of the corresponding material states are taken as the hyper-parameters. Arc-length solver has been used.

The results obtained for the damage profile at the center section of the bar in Figure 16 have been presented in Figure 11 and the load-displacement response in Figure 12.



**Figure 11:** Damage distribution without regularization. (a) Damaged elastic modulus used for hyper-parameter and no AL, (b) Tangent modulus used for hyper-parameter and no AL, (c) Tangent modulus used for hyper-parameter with AL solver, (d) Fine mesh - Tangent modulus used for hyper-parameter and no AL, (e) Fine mesh - Tangent modulus used for hyper-parameter with AL solver



**Figure 12:** Load - displacement response for the bar with a notch. AL stands for Arc Length. Elastic - no AL corresponds to case (a), Tangent - no AL corresponds to case (b), AL corresponds to case (c), AL - fine corresponds to case (e) of Figure 11.

When the damaged elastic moduli are taken as the hyper-parameter, the damage profile is diffuse as presented in Figure 11a, even if there is no regularization in the problem. This is inline with the earlier discussion regarding the path taken by the mechanical and the material states in 1D, presented in Figures 3a and 7a, where the direction of strain increment in an element may result in a transition towards a homogeneous solution branch. The corresponding

load-displacement curve in Figure 12 is still almost straight when the simulation is terminated. The simulation has been terminated early since the number of iterations till convergence (till successive material points are close) are very high. The distance functional remains large even though the successive material iterates are closer.

When the tangent moduli are taken as the hyper-parameter, the damage profile initially localizes until a certain load step, as presented in Figure 11b. After this, a snapback response is expected. Since the system is displacement driven, it is expected that the system jumps through and the fully damaged band develops over the entire section instantaneously in a single load step. However, as presented in Figure 11b, after the initial localized profile, the damage becomes diffuse as in the earlier case, and also as presented in Figure 7b. The corresponding load-displacement curve in Figure 12 reveals a similar behavior. At the point where a snapback is expected, there is a small drop in the load, which later on recovers and starts to increase again.

The next combination is the arc-length solver with the tangent moduli as the hyper-parameter. Like in the 1D case, the damage localizes to a one element wide region as shown in Figure 11c. The arc length solver is able to track the severe snapback response of the bar as presented in Figure 12. This also demonstrates the robustness of the arc-length constraint, which will be described in detail later.

When similar combinations are tested for a fine mesh, the elastic-no arc length combination needed an extremely large number of iterations to converge, at least once the damage begins to increase. This is clear from the analysis in the previous section, where the spectral radius of the matrix  $\mathbf{AC}$  gets much closer to 1 as the mesh is refined. The tangent-no arc length combination converges at an optimal rate until the snapback instability occurs. At that point, it fails to converge at all. The resulting damage profile from Figure 11d, however, resembles the one obtained by using an arc-length solver in Figure 11e. The load-displacement curve for this case has not been presented, since the curve is almost linear and shows no special features till the point of unconvexity. The tangent-arc length combination presented in Figure 11e presents a damage band that is one-element wide, exhibiting the expected mesh sensitivity. The load-displacement curve, presented in Figure 12, shows a snapback response that is different from the coarse mesh case.

**Remark.** *Results from the analysis performed in A reveals that an optimal rate of convergence is obtained when the metric in the distance functional is taken equal to the tangent,  $\frac{\partial \sigma^*}{\partial \epsilon^*}$ , obtained from the corresponding material state. For materials that undergo damage, this derivative is negative. Thus, during the alternate minimization process, for the elements that undergo damage, C is smaller than zero. It shall be noted that the convergence of the solver is still judged on the basis of the metric that is equal to the elastic modulus and the tangent is used only during the alternate minimization and the corresponding Newton iterations, equations 6 and 9.*

In a non-DD setting, the problem involves minimizing a functional that is non-convex. The energy density depends on displacement and damage, and is separately convex with respect to each variable. Consequently, the minimization is carried out by alternating between the two variables, which effectively convexifies the problem. This procedure allows the minimizers to jump through instabilities such as snapbacks under displacement control (although this may require a larger number of iterations). It should be noted that using a monolithic (Newton) solver for such a non-convex problem, without additional refinements or an arc-length constraint, may lead to non-convergence [13, 15, 32, 2]. In the DD setting, however, the database resembles that in figure 1, and thus the non-convexity is inherently embedded in the stored data. For this reason, an arc-length method has been employed to trace the entire bifurcation branch without requiring any jumps. As presented, not using an arc-length solver may result in a widened damage profile even in the absence of regularization. This may result in a mesh-dependent behavior even after the introduction of a length scale.

To conclude, with the ingredients presented (the arc length solver and the choice of hyper-parameters), the data driven solver is able to exhibit the expected mesh dependence of the solution as expected when there is no regularization. To emphasize, the emergence of localized or diffuse damage in this setting is not enforced but results entirely from the interaction between the DD hyper-parameters, continuation strategy, and material non-convexity.

#### 4. Introduction of a length scale

In the previous section, it was shown that, in the absence of regularization, the DD algorithm exhibits mesh dependence of the solution, as expected, provided that the appropriate algorithmic ingredients are employed. The natural next step is therefore to introduce a regularization mechanism such that the results, most notably the width of the localization band, become independent of the mesh size.

In non-DD settings, this objective is commonly achieved through a variety of approaches, including gradient-enhanced damage models, nonlocal damage evolution laws, and nonlocal or gradient-based strain formulations. In the DD setting considered here, however, the direct use of damage gradients is non-trivial or even infeasible, since the damage variable, when present, is attached to the material state rather than defined as an independent field. For this reason, regularization is instead introduced through the gradient of the mechanical strain.

Furthermore, the regularization is modeled independently of the material database - here represented by the local constitutive model - in contrast to the approach adopted in [33].

To this end, the strain-based Lipschitz regularization introduced in [16] is first adapted to the data-driven framework in a one-dimensional setting. The method is then applied to the localization problem introduced in the previous section. Finally, a relaxed version of the regularization strategy is developed and presented for the two-dimensional case.

#### 4.1. The method

The regularization technique was introduced in [16], where it was tested with a constitutive model. Here, it will first be extended for use with the DDCM methodology. To begin with, a length scale is introduced into the problem by restricting the gradient of the strain variable to lie in a certain interval. More precisely, it is required that  $\epsilon_{,x} \in [-1/\ell_c, 1/\ell_c]$ , where  $\ell_c$  is a length parameter. It is expected that the introduction of this constraint prevents the spurious strain localization and the dependence of the solution on the Finite Element mesh size.

The introduction of the Lipschitz constraint on the strain variable modifies the equation of equilibrium in 1D to

$$\sigma_{,x} - \lambda_{,xx} = 0, \quad (17)$$

see equation 37 of [16], where  $\lambda$  is the Lagrange multiplier that is dual to the Lipschitz constraint.  $\lambda \neq 0$  only in the regions of the body where the constraint is active. The equations of equilibrium used in DDCM should thus take this into account.

Assuming piecewise affine interpolation for the displacement variable and thus, a piecewise constant variation of the stresses and the strains, the distance functional becomes

$$d(\epsilon, \sigma, \epsilon^*, \sigma^*) = \frac{1}{2}(\mathbf{B}\mathbf{u} - \epsilon^*)^T C\mathbf{w}(\mathbf{B}\mathbf{u} - \epsilon^*) + \frac{1}{2}(\sigma - \sigma^*)^T C^{-1}\mathbf{w}(\sigma - \sigma^*). \quad (18)$$

The objective now is to find the minimizers of the above functional so that the strain field satisfies the Lipschitz constraint, and the stress fields satisfy the equilibrium. The Lagrange multipliers that enforce the constraint on the strain field act as couple stresses. The strains are interpolated on a different mesh from the one used to interpolate the displacement field. See Figure 13. The strain field is taken to vary linearly over the red mesh. The solution is now

$$\text{Sol} \in \arg \min_{(\epsilon, \sigma) \in \mathcal{E}, (\epsilon^*, \sigma^*) \in \mathcal{D}} d(\epsilon, \sigma, \epsilon^*, \sigma^*), \quad (19)$$

where, the equilibrium manifold is now defined as (in discrete form)

$$\mathcal{E} = \{(\epsilon, \sigma) : \epsilon = \mathbf{B}\mathbf{u}, \epsilon \text{ is such that } |\mathbf{B}'\epsilon| \leq 1/\ell_c, \mathbf{u} = \mathbf{u}_D \text{ on } \partial\Omega_d, \text{ and } \mathbf{B}^T \mathbf{w}\sigma + \mathbf{P}^T \lambda = 0\}, \quad (20)$$

where  $\lambda$  is the Lagrange multiplier.  $\mathbf{B}'$  is the discrete gradient operator on the strain variable (see Figure 13), and  $\mathbf{P} = \mathbf{B}\mathbf{B}'$  is the discrete second gradient operator. By splitting the constraints  $|\mathbf{B}'\epsilon| \leq 1/\ell_c$  as  $\mathbf{B}'\epsilon \leq 1/\ell_c$  (plus side) and  $-\mathbf{B}'\epsilon \leq 1/\ell_c$  (minus side), the Lagrangian can now be defined as:

$$\begin{aligned} \Pi = d(\epsilon, \sigma, \epsilon^*, \sigma^*) - \eta^T \mathbf{B}^T \mathbf{w}\sigma + \sum_{i=1}^{N_c^+} \lambda_i^u \left( (\mathbf{P}\mathbf{u})_i - \frac{1}{\ell_c} \right) + \sum_{i=1}^{N_c^+} \lambda_i^{\eta u} \left( (\mathbf{P}(\mathbf{u} + \eta))_i - \frac{1}{\ell_c} \right) + \\ \sum_{i=1}^{N_c^-} \lambda_i^u \left( -(\mathbf{P}\mathbf{u})_i - \frac{1}{\ell_c} \right) + \sum_{i=1}^{N_c^-} \lambda_i^{\eta u} \left( -(\mathbf{P}(\mathbf{u} + \eta))_i - \frac{1}{\ell_c} \right). \end{aligned} \quad (21)$$

Here,  $N_c^+$  and  $N_c^-$  denote the number of active plus and minus constraints, respectively, and  $\lambda_i^u$  are the Lagrange multipliers dual to the constraint on the gradient of strain, and  $(\bullet)_i$  denotes the  $i$ th component of the vector  $\bullet$ . The solution is defined as the collection that renders the Lagrangian stationary.

$$\Pi \leftarrow \text{stat!} \quad (22)$$

The third and the fourth terms in the Lagrangian warrant some discussion. It is clear that the third term, together with the condition that  $\lambda^u \geq 0$ , enforces the inequality constraint  $|\epsilon_{,x}| \leq 1/\ell_c$ . Treating  $\boldsymbol{\eta}$  as the virtual displacement, the fourth term enforces the constraint that in the regions where the inequality constraint is satisfied as an equality, the perturbed field,  $\mathbf{u} + \boldsymbol{\eta}$ , must also satisfy the constraint as an equality. It shall be noted that there is a priori no restriction on the sign of  $\lambda_{\eta u}$ . However, as will be demonstrated below, at convergence,  $\lambda_i^{\eta u} = -\lambda_i^u$ . This, owing to the restriction on  $\lambda_u$ , determines the sign of  $\lambda_{\eta u}$ .

For notational convenience, the minus constraints will be dropped, since the only difference they make is in the sign of the couple stress. The Euler Lagrange equations can be seen to be

$$\mathbf{r}_u = \mathbf{B}^T C \mathbf{w} \mathbf{B} \mathbf{u} - \mathbf{B}^T C \mathbf{w} \epsilon^* + \mathbf{P}^T \lambda^u + \mathbf{P}^T \lambda^{\eta u} = 0, \quad (23)$$

$$\mathbf{r}_{\eta} = -\mathbf{B}^T \mathbf{w} C \mathbf{B} \boldsymbol{\eta} - \mathbf{B}^T \mathbf{w} \sigma^* + \mathbf{P}^T \lambda^{\eta u} = 0, \quad (24)$$

$$r_{\lambda_i^u} = (\mathbf{P} \mathbf{u})_i - \frac{1}{\ell_c} = 0, \quad (25)$$

$$r_{\lambda_i^{\eta u}} = (\mathbf{P}(\mathbf{u} + \boldsymbol{\eta}))_i - \frac{1}{\ell_c} = 0. \quad (26)$$

At convergence,  $\epsilon = \epsilon^*$  and  $\boldsymbol{\eta} = 0$ . Thus, from equation 23, it can be seen that  $\lambda_i^u = -\lambda_i^{\eta u}$ , as mentioned earlier. The equations of equilibrium can be recovered as  $\mathbf{B}^T \mathbf{w} \sigma + \mathbf{P}^T \lambda^u = 0$ , since  $\sigma = \sigma^*$  at convergence.

---

**Algorithm 1** The DD algorithm

---

Given  $\sigma^n, \epsilon^n, \sigma^{*n}, \epsilon^{*n}, \Delta\epsilon$   
Determine  $\sigma^{n+1}, \epsilon^{n+1}, \sigma^{*n+1}, \epsilon^{*n+1}$   
 $i \leftarrow 1, \sigma^*_0 = \sigma^{*n}, \epsilon^*_0 = \epsilon^{*n}$   
**while** Not Converged **do**  
  **procedure** MECHANICAL UPDATE( $\sigma^*_{i-1}, \epsilon^*_{i-1}$ , Tangent)  
    Resolve the Euler Lagrange equations  
    **return**  $\sigma_i, \epsilon_i$   
  **end procedure**  
  **procedure** MATERIAL UPDATE( $\sigma_i, \epsilon_i$ )  
    Find the closest point on the elastic branch, equation 47:  $\sigma^*_e, \epsilon^*_e$   
    Find the closest point on the damage branch, equation 46:  $\sigma^*_d, \epsilon^*_d$   
    **if**  $d(\sigma_i, \epsilon_i, \sigma^*_e, \epsilon^*_e) < d(\sigma_i, \epsilon_i, \sigma^*_d, \epsilon^*_d)$  **then**  
      **return**  $\sigma^*_e, \epsilon^*_e$ , Elastic degraded modulus  
    **else**  
      **return**  $\sigma^*_e, \epsilon^*_e$ , Tangent modulus  $\left( \frac{\partial \sigma^*}{\partial \epsilon^*} \right)$   
    **end if**  
  **end procedure**  
   $i \leftarrow i + 1$   
**end while**  
 $n \leftarrow n + 1$

---

## 4.2. Application and results

The regularization technique is now applied to the DDCM case tested earlier with a body force. The value of  $\ell_c$  is taken to be 0.1m. Arc length solver has been used like in the previous case with similar loading conditions. A piecewise linear and continuous interpolation has been used for displacement and a piecewise constant for the mechanical stress. A dual mesh had been used to impose the constraint on the strain gradient. The strain is taken to be piecewise linear and continuous on the dual mesh. See Figure 13. In this case, the discrete gradient operator  $\mathbf{B}'$  is calculated on the dual (red) mesh.

The load displacement curves and the strain distribution at failure have been presented in Figures 14a and 14b for two different mesh sizes (compare with Figure 10). A couple of observations warrant a closer look. First, as discussed

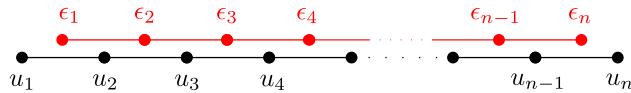
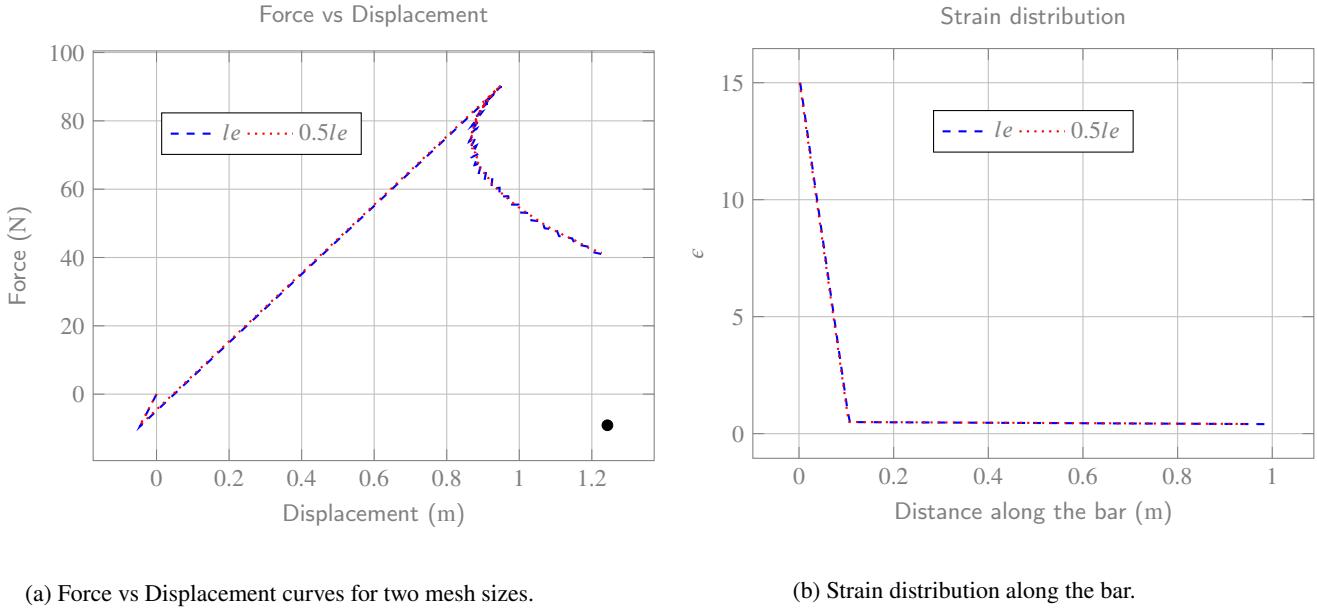


Figure 13: Mesh for displacement and strain



(a) Force vs Displacement curves for two mesh sizes.

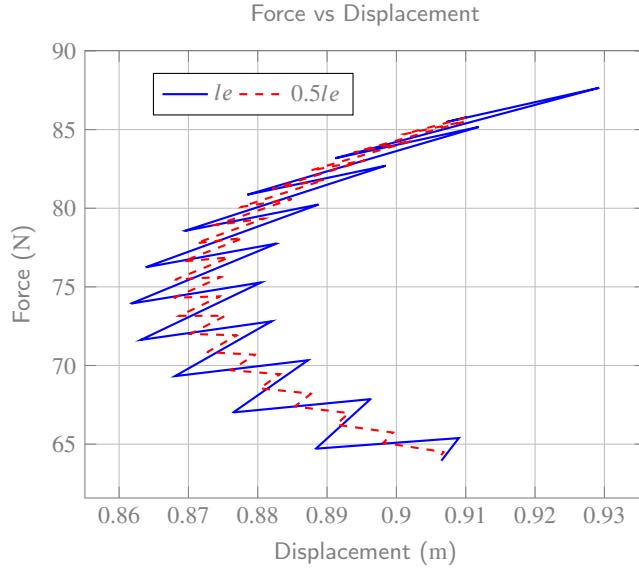
(b) Strain distribution along the bar.

Figure 14: Results with regularization

in [16], the constraint shall be removed from the elements that are fully damaged to allow for the strains to localize in the fully damaged element and prevent the expansion of damage further into the body. The black dot in Figure 14a corresponds to the case when the constraint is deactivated - the force displacement curve suddenly jumps to zero force. This situation is trivial in 1D, but is susceptible to cause convergence issues in 2D. See [14, 30, 22] and the references therein for similar discussions where the damaged regions extend across the body and various modifications/adaptations of the corresponding regularization method have been proposed to force a localized damage profile. Second, taking a closer look at the force-displacement curve as in figure 15, a number of smaller snapbacks can be observed. It can be explained as follows: since the regularization is introduced through an inequality constraint, and this constraint is numerically resolved with a finite mesh size, the regularization is inactive until strain localizes in some region of the body. The strains then grow locally till the constraint becomes active. Once activated, the constraint expands the damage zone further into the neighboring elements, where the strains again grow locally. This process carries on until some part of the body is fully damaged. Thus, the problem switches between a local (albeit with a varying length scale) and non-local problem as the constraint is activated and the damage spreads into other elements. The snapback occurs when the strains localize and the recovery corresponds to the activation of the constraint. It can also be seen that the amplitude of this sawtooth like pattern decreases as the mesh is refined, but the number of tooth increases.

**Remark.** *In hindsight, deactivating the constraint only after an element is fully damaged does indeed cause convergence issues in 2D. This issue has been discussed in multiple references - in the context of non-local damage models in [14]. Thus, for the 2D case, the constraint is gradually weakened as the damage value increases. Also, to reduce the impact of sawtooth like pattern, the inequality constraint is weakened and a log-barrier-like function is used instead. More details can be found in the sections below.*

Thus, in the 1D case, the proposed regularization technique ensures that the result remains independent of the mesh size, as expected. Though not reported here, the simulations performed in 2D case reveal convergence issues if the constraint is abruptly deactivated once an element fails completely ( $d = 1$ ). Also, the activation-deactivation of elements lead to sawtooth like pattern in the load displacement curve, which may prove tedious to trace in 2D. To



**Figure 15:** Force vs Displacement curves for two mesh sizes zoomed in.

address these challenges, an alternative approach is presented below for the 2D case, in which the regularization is gradually weakened as the damage increases.

## 5. 2D Case

As mentioned in the previous section, the strain based Lipschitz constraint introduces a length scale into the problem and results in a mesh independent solution. However, as was seen, the inequality constraint introduces a saw-tooth like pattern in the force-displacement response, which can be tedious to track. Also, to ensure that the damage zone does not extend across the body, the constraint (and thus, the length scale) will have to be removed from the fully damaged regions of the body. Abrupt removal of the constraint will likely lead to issues with convergence as well. Thus, two modifications will be introduced in this section in its extension to 2D. First, the constraint will be gradually weakened as the damage level increases. Second, the inequality constraint will be replaced by a log-barrier-like function. It is expected that the log-barrier like regularization will result in a smaller number of peaks in the saw-tooth pattern reported in the 1D case, if not completely eliminate it. It is also expected that the log-barrier like form delays the effect of the regularization term until the strains start to localize and not become active right from the earlier stages before strain localization occurs. See Figure 23.

The body is assumed to be triangulated by constant strain triangles, referred to as the primary mesh. Consequently, the displacement is piecewise linear and continuous across the elements, while the stress field is piecewise constant in each element. Thus, each element is associated with a single mechanical and material state. To impose the constraints on the gradient of strain, a dual mesh is introduced. The dual mesh consists of another set of constant strain triangles defined using the centroids of the triangles of the primary mesh as vertices. The strain field is then interpolated over the dual mesh in a piecewise linear manner. The gradient of strain, is thus piecewise constant over the triangles of the dual mesh.

The distance functional remains as

$$d(\boldsymbol{\epsilon}, \boldsymbol{\sigma}, \boldsymbol{\epsilon}^*, \boldsymbol{\sigma}^*) = \frac{1}{2} (\mathbf{B}\mathbf{u} - \boldsymbol{\epsilon}^*)^T \mathbf{C} \mathbf{w} (\mathbf{B}\mathbf{u} - \boldsymbol{\epsilon}^*) + \frac{1}{2} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*)^T \mathbf{C}^{-1} \mathbf{w} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^*). \quad (27)$$

Similar to the 1D case, the solution is then defined as the one that renders the following Lagrangian stationary.

$$\Pi = d - \boldsymbol{\eta}^T \mathbf{B}^T \mathbf{w} \boldsymbol{\sigma} - \mu \sum_{i=1}^{N_c} w'_i \ln s_{ui} + \mu \sum_{i=1}^{N_c} w'_i \ln s_{\eta i} + \sum_{i=1}^{N_c} \lambda_u^i \left\{ s_{ui} - \left[ \frac{1}{\ell_c} - \psi_i(\mathbf{u}) \right] \right\} + \sum_{i=1}^{N_c} \lambda_\eta^i \left\{ s_{\eta i} - \left[ \frac{1}{\ell_c} - \psi_i(\mathbf{u} + \boldsymbol{\eta}) \right] \right\}. \quad (28)$$

The function  $\psi$  is the strain-gradient term, as defined in equation 2.3 of [24]. In the current case, it is defined as

$$\psi = 3k_{11i}k_{11i} + 3k_{22i}k_{22i} + \frac{1}{2}k_{12i}k_{12i}, \quad (29)$$

where  $k_{lmn} = \epsilon_{lm,n}$  and Einstein summation convention has been used. A slack variables  $s \geq 0$  have been introduced. Using Lagrange multipliers  $\lambda_u$ , the slack variables are defined as the difference between the strain gradient term and the strain gradient term. The log term stays small as the slack variables remain small and begins to take over as the constraint is approached. This introduces the length scale into the problem. Inspired from the 1D case as in equation 21, a term involving the 'virtual displacements'  $\eta$  is also included in the above equation in the strain gradient term  $\psi$  in addition to the one involving displacement. By taking variations with respect to  $\eta$ , equilibrium equation can be obtained, which consists of the Cauchy stress  $\sigma$  and the Lagrange multipliers  $\lambda_{\eta u}$  that act as couple stress. The modified equilibrium equation because of the strain gradients is thus taken care of automatically.

The displacement field satisfies Dirichlet boundary conditions -  $\mathbf{u} = \mathbf{u}_D$  on  $\partial\Omega_D$ . The test function  $\eta$  is equal to  $\mathbf{0}$  on  $\partial\Omega_D$ . The factors  $w'_i$  denotes the area of the dual elements and is thus similar to  $w$  in equation 3.  $s_{ui}, s_{\eta i}$  are the slack variables.  $\lambda^i$  are the Lagrange multipliers that define the slack variables.  $\psi_i$  are the energies associated with the gradient of strain as defined below. When the constraints are to be relaxed gradually, the Lagrangian becomes

$$\Pi = d - \eta^T \mathbf{B}^T \mathbf{w} \sigma - \mu \sum_{i=1}^{N_c} w'_i \ln s_{ui} + \mu \sum_{i=1}^{N_c} w'_i \ln s_{\eta i} + \sum_{i=1}^{N_c} \lambda_u^i \left\{ s_{ui} - \left[ \frac{1}{\ell_c} - g(d_i) \psi_i(\mathbf{u}) \right] \right\} \\ + \sum_{i=1}^{N_c} \lambda_{\eta}^i \left\{ s_{\eta i} - \left[ \frac{1}{\ell_c} - g(d_i) \psi_i(\mathbf{u} + \eta) \right] \right\}, \quad (30)$$

where  $g$  is a function such that  $g(0) = 1$  and  $g(1) = 0$ . The function  $\psi$  from equation 29 can be written in discrete form over a dual element  $i$  as

$$\psi_i(\mathbf{u}) = \mathbf{u}^T \mathbf{G}_i^T \mathbf{G}_i \mathbf{u}, \quad (31)$$

where  $\mathbf{G}_i$  denotes the discrete second gradient operator, written on the  $i^{th}$  dual element.

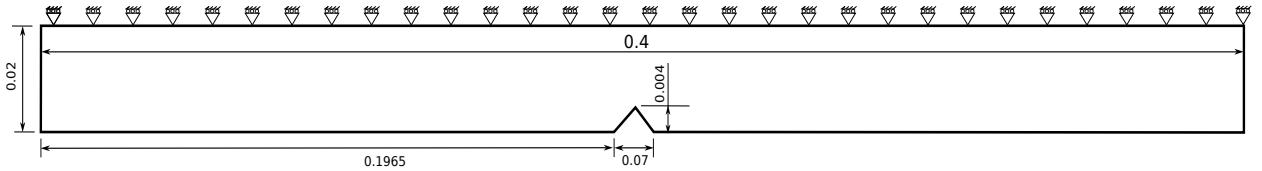
**Remark.** A couple of observations are warranted. First, the function  $g$  is selected so that  $g(0) = 1$  and  $g(1) = 0$ . This means that when there is no damage, the regularization is fully effective, and when fully damaged, the regularization is removed. The transition between the two states is gradual if  $g$  is smooth. Second, in this study, the parameter  $\mu$  is fixed to be a finite positive value. It shall be noted that the method resembles the interior point method of resolving inequality constraints in the standard optimization setting, where successive problems are solved with decreasing  $\mu$ ,  $\mu \rightarrow 0$ . Thus, as the parameter  $\mu$  is made smaller, the method presented here boils down to the inequality case and the problems/features about snapback that were observed in the 1D case return. It is expected that taking  $\mu$  to be finite and fixed prevents this issue.

The displacement vector  $\mathbf{u}$  is split into degrees of freedom that are unknown or free,  $\mathbf{u}_f$ , and the part that follows from the applied boundary conditions,  $\mathbf{u}_g$ . Specifically, if  $\mathbf{u} \in \mathbb{R}^N$ ,  $\mathbf{u}_g \in \mathbb{R}^N$  is a vector of zeros and ones, with ones at places corresponding to the nodes where the boundary conditions are applied. The vector  $\mathbf{u}_f \in \mathbb{R}^{N_f}$  will be determined from the solution of the problem. Thus,  $\mathbf{u} = \mathbf{C}_f \mathbf{u}_f + \lambda_g \mathbf{u}_g$ , where  $\mathbf{C}_f \in \mathbb{R}^{N \times N_f}$  is a matrix that relates the total and the free degrees of freedom,  $\lambda_g \in \mathbb{R}$  is the strength of the applied boundary condition on  $\partial\Omega_D$ . Since  $\eta = \mathbf{0}$  on  $\partial\Omega_D$ , it is simply expressed as  $\eta = \mathbf{C}_f \eta_f$ . Using the expressions for  $\mathbf{u}$  and  $\eta$  in the Lagrangian and taking its variations results in the Euler Lagrange equations.

$$\mathbf{r}_{\mathbf{u}_f} = \mathbf{C}_f \mathbf{B}^T \mathbf{C} \mathbf{w} (\mathbf{B} \mathbf{u} - \mathbf{\epsilon}^*) + \sum_{i=1}^{N_c} 2\lambda_u^i g(d_i) \mathbf{C}_f \mathbf{G}_i^T \mathbf{G}_i \mathbf{u} + \sum_{i=1}^{N_c} 2\lambda_{\eta}^i g(d_i) \mathbf{C}_f \mathbf{G}_i^T \mathbf{G}_i (\mathbf{u} + \eta) = \mathbf{0}, \quad (32)$$

$$\mathbf{r}_{\eta_f} = -\mathbf{C}_f \mathbf{B}^T \mathbf{w} (\sigma^* + \mathcal{C} \mathbf{B} \eta) + \sum_{i=1}^{N_c} 2\lambda_{\eta}^i g(d_i) \mathbf{C}_f \mathbf{G}_i^T \mathbf{G}_i (\mathbf{u} + \eta) = \mathbf{0}, \quad (33)$$

$$r_{\lambda_u^i} = \mathbf{u}^T \mathbf{G}_i^T \mathbf{G}_i \mathbf{u} - \frac{1}{\ell_c} + s_{ui} = 0, \quad (34)$$



**Figure 16:** Geometry of the bar with a notch. All dimensions are in meters.

$$r_{\lambda_\eta^i} = (\mathbf{u} + \boldsymbol{\eta})^T \mathbf{G}_i^T \mathbf{G}_i (\mathbf{u} + \boldsymbol{\eta}) - \frac{1}{\ell_c} + s_{\eta i} = 0, \quad (35)$$

$$r_{s_{ui}} = \lambda_u^i s_{ui} - \mu w'_i = 0, \quad (36)$$

$$r_{s_{\eta i}} = \lambda_\eta^i s_{\eta i} - \mu w'_i = 0. \quad (37)$$

The above system is resolved for  $\mathbf{u}_f, \boldsymbol{\eta}_f, \lambda_u, \lambda_\eta, s_u, s_\eta$ , knowing  $\lambda_g$ . When arc length solver is used,  $\lambda_g$  is treated as an unknown and the system is supplemented by the arc-length equation. The tangent matrix used during the Newton iterations can be seen in the appendix. The details of the arc-length solver can be found in D.

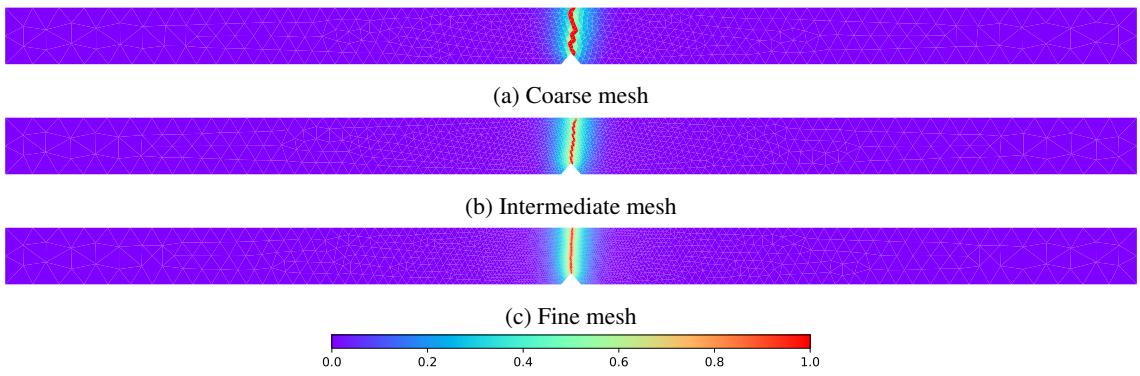
**Remark.** As in the 1D case, at convergence,  $\epsilon = \epsilon^*$ , and  $\sigma = \sigma^*$ . Thus,  $\lambda_\eta^i = -\lambda_u^i$ . Also, the Lagrange multipliers that take the role of couple stresses are computed as  $\lambda_u^i = \mu w'_i / s_{ui} > 0$  since  $s_{ui} > 0$  as a consequence of the log-barrier-like regularization.

## 6. Applications

The formulation presented above will be applied for the case of a bar with a notch and an L-beam. The intent is to examine the effectiveness of the regularization technique in terms of making the solution mesh independent.

### 6.1. A bar with a notch

The regularization technique described above has been applied to a notched bar problem, as illustrated in Figure 16. The left end of the bar is fixed, top end of the bar is constrained to move horizontally, and a horizontal displacement has been applied at the right end. The displacement magnitude is controlled using the arc-length method, which is detailed in the Appendix. The displacement field is discretized using constant-strain triangular elements, as described earlier. The strain variable  $\mathbf{s}$  is subsequently interpolated onto a dual mesh, which is then used to evaluate the strain gradient.

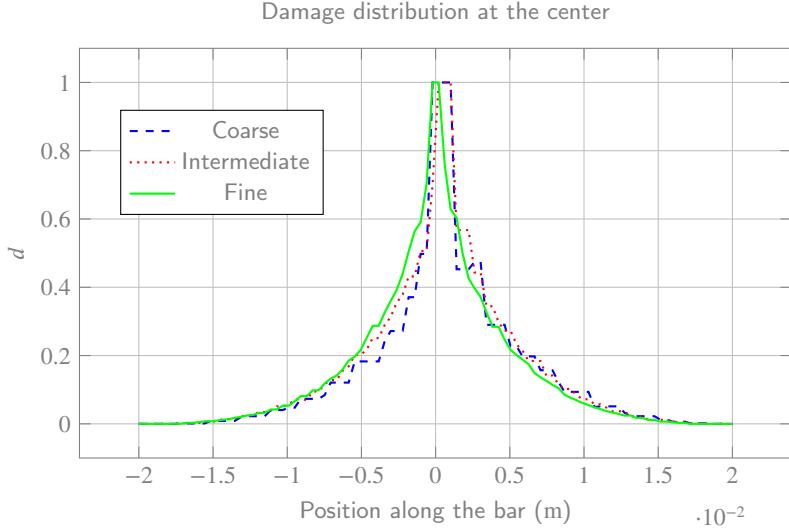


**Figure 17:** Results with regularization - A notched bar

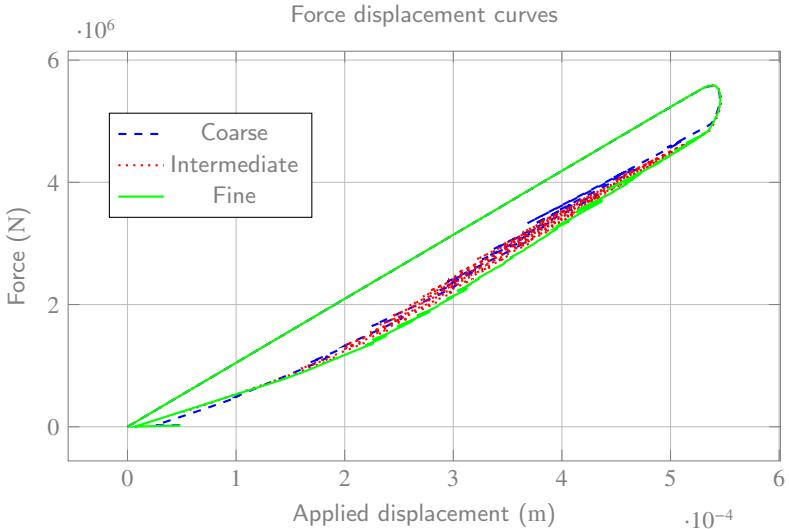
The geometry of the bar is shown in Figure 16. The material parameters used for this study were  $E = 210E9$ ,  $\nu = 0.3$ ,  $Y_c = 2E6$ ,  $k = 10$ ,  $\ell_c = 0.5$ , see C. The parameter  $\mu = 2E6$  in equation 30. The function  $g$  is taken same as in equation 49.

Figure 17 shows the damage distribution attached to the material states associated with each element. The results are for three mesh sizes, where the ratio of element sizes of the coarse mesh to the intermediate mesh, and of the

intermediate mesh to the fine mesh is 2. Compared with Figure 11, the damage profile is spread over multiple elements and is mesh independent. The variation of damage value across the center of the bar, from  $(0.08, 0.01)$  to  $(0.12, 0.01)$ , where the origin is at the bottom left corner of the bar, can be seen in figure 18. Force-displacement curves for all the three mesh sizes can be seen in figure 19. The results are different from the ones presented in Figure 12 without any regularization.

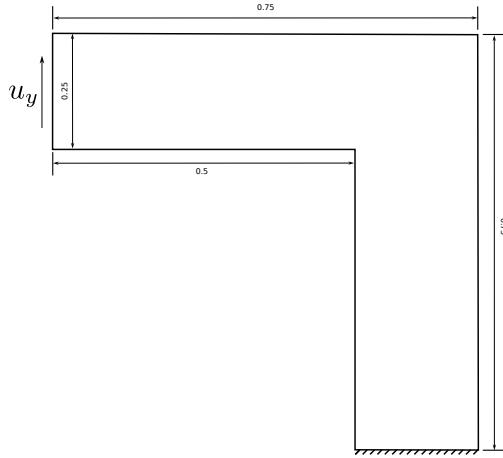


**Figure 18:** Variation of damage along the center of the bar,  $y = 0$ ,  $x \in [-0.02, 0.02]$  m.



**Figure 19:** Force displacement curves of the 2D-1D bar for three different meshes.

It can be seen that in all the cases, the results are independent of the mesh size. The width of the damage zone does not change as the mesh is refined. Likewise, the force-displacement curve also remains the same for all the mesh sizes. A severe snap-back can be seen in the figure 19, which has been captured by the arc length solver. It shall be noted that the saw-tooth-like pattern that has been observed in the 1D case, where the constraint is strictly enforced, is again observed even in the current scenario. Perhaps testing with different values of  $\mu$  in the solver or using a different



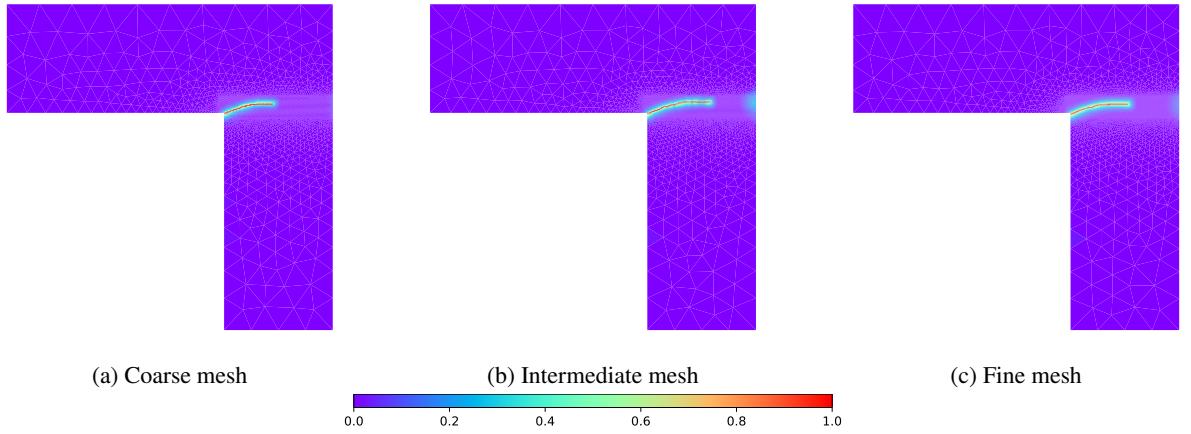
**Figure 20:** Geometry of the L-beam. All dimensions are in meters.

arc length solver might smoothen the results. This has not been pursued further in this article and can be the focus of future work.

## 6.2. L Beam

The regularization technique has also been applied to an L-shaped beam (L-beam) problem. This example allows to examine the regularization effects when crack path is curved. The horizontal face of the beam is fully constrained, enforcing  $u_x, u_y = 0$ , while a prescribed vertical displacement was applied on the vertical face. See Figure 20. The magnitude of this displacement is controlled using the arc-length method to accurately capture the nonlinear response and post-peak behavior.

The material parameters are the same as the previous example except  $\ell_c = 1$ . A crack is expected to initiate at the inner corner of the L-beam due to the pronounced stress concentration in this region.

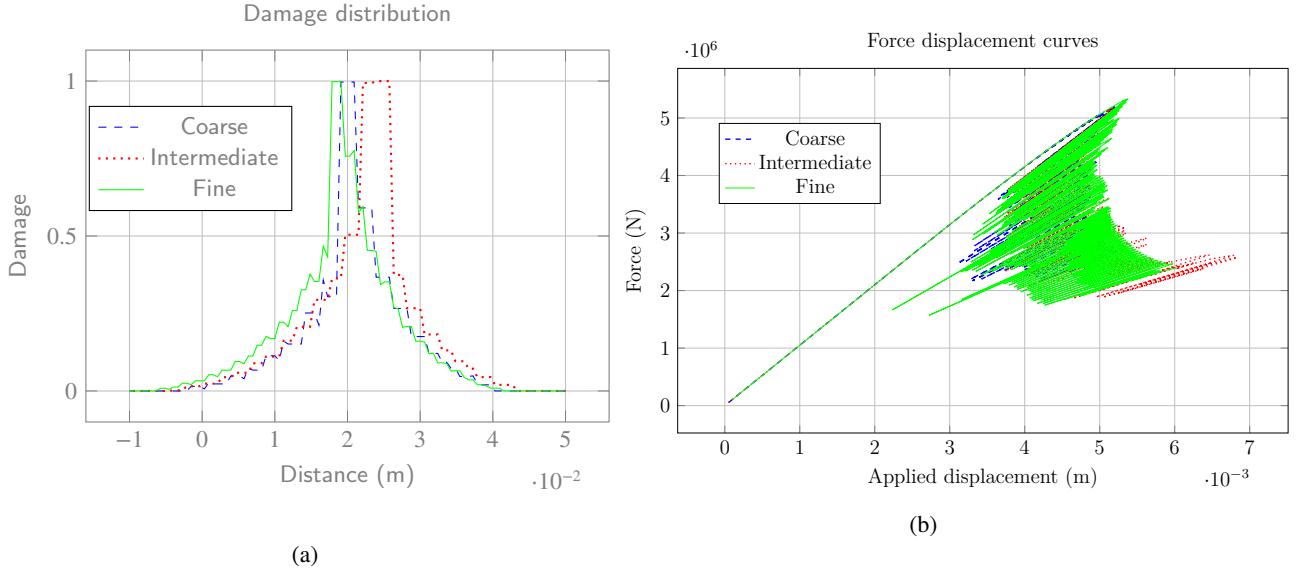


**Figure 21:** Results with regularization

Simulations have been performed using coarse, intermediate, and fine meshes. The corresponding characteristic element sizes in the damaged region, computed as  $4A/\sqrt{3}$ , where  $A$  is the area of the triangle in that region, are 0.0056 m, 0.0048 m, 0.0036 m for the coarse, intermediate, and fine mesh, respectively. The resulting damage fields for the three discretizations are shown in Figure 21.

To further examine mesh sensitivity, the variation of the damage variable along a vertical section at  $x = 0.08$ ,  $y \in [-0.01, 0.05]$  is plotted in Figure 22a. The width of the damage zone remains nearly constant across all meshes,

indicating satisfactory mesh objectivity. The position of the maximum damage varies slightly, with the peaks for the intermediate and fine meshes differing by approximately 0.005 m. The force-displacement responses for all three mesh levels are compared in Figure 22b. The overall behavior is consistent across discretizations, confirming convergence of the global response. Nevertheless, the characteristic sawtooth pattern observed previously in the one-dimensional test case reappears, despite the use of the interior-point regularization to relax the strict inequality constraint. This behavior suggests that, while the proposed formulation improves numerical stability, some degree of non-smoothness still remains in the progressive damage evolution.



**Figure 22:** (a) Variation of damage along,  $x = 0.85$ ,  $y \in [-0.01, 0.05]$  m. (b) Force displacement curves of the 2D-1D bar for three different meshes.

## 7. Discussion

The present study employs a constitutive model for the material update within the data-driven computational mechanics framework. **This choice corresponds to the limit of an infinitely dense material database and is adopted to isolate the algorithmic behavior of the alternating minimization scheme from database-resolution effects.** In the presence of damage-induced non-convexity, these two sources of difficulty can interact in a manner that obscures the intrinsic properties of the solver. By considering the infinite-database limit, the present study provides a controlled setting in which convergence mechanisms, instabilities, and the role of additional algorithmic ingredients can be examined independently.

The results demonstrate that, in problems involving material softening, the choice of hyper-parameters and continuation strategy in data-driven solvers influences not only convergence rates but also the physical character of damage localization. In particular, the combination of an arc-length solver with tangent-based hyper-parameters is shown to be essential for correctly capturing localization and snap-back behavior. This choice yields robust convergence properties that are insensitive to mesh refinement, while alternative combinations may lead to diffuse damage patterns or a complete loss of convergence as the mesh is refined. Tangent-based hyper-parameters may not always be positive definite. Thus, these set of hyper-parameters may be reserved for the mechanical update process in the corresponding Euler-Lagrange equations and a positive definite secant modulus can be used for hyper-parameters during the material update. This may lead to a loss of rate of convergence, but preserve the instability tracking ability of the hyper-parameter - arc-length combination while dealing with discrete databases.

The need for algorithmic modifications in the presence of instabilities has been recognized previously in the data-driven literature. Kuang et al. [20], for example, considered instabilities arising from geometric nonlinearities and demonstrated that appropriate modifications to the solution strategy are required to correctly trace snap-back

responses. In an analogous manner, the present study shows that employing the local material tangent as a hyper-parameter allows the data-driven equilibrium operator to develop zero eigenvalues when a snap-back response is expected. When combined with an arc-length solver, this modification enables the data-driven formulation to capture material instabilities in a manner consistent with softening behavior.

Tangent-based hyper-parameters have previously been explored in data-driven methods primarily as a means of accelerating convergence, and only in settings where instabilities are absent [29, 25, 17]. To the best of the authors' knowledge, the present work is the first to demonstrate that the choice of hyper-parameters, together with an appropriate continuation strategy, plays a critical role in the treatment of material instabilities within a data-driven framework.

The observed behavior can be interpreted in close analogy with finite element formulations employing constitutive models. In such settings, non-convexity arising from material softening is known to cause standard Newton iterations to fail, motivating the use of arc-length solvers and consistent material tangents to robustly trace snap-back responses. In this context, the present data-driven formulation using an arc-length solver together with tangent-based hyper-parameters may be viewed as the analogue of a monolithic Newton method equipped with a consistent material tangent. Also, the regularization method introduced relies on the gradient of the strain variable, and the regularization is removed once the damage variable reaches 1. [23] analyzes the effectiveness of regularization in such cases and concludes that the regularization in such cases is ineffective and the damage evolution is discontinuous. Coupling the method with an arc-length solver, however, forces the damage evolution to be continuous and is likely to keep the regularization effective, as demonstrated.

Having established the algorithmic requirements for handling instabilities, the role of regularization becomes central in ensuring mesh-independent responses. Incorporating strain-based Lipschitz regularization within the data-driven framework yields mesh-independent damage profiles in one and two dimensions. At the same time, the results highlight practical challenges associated with inequality-based regularization, including oscillatory force-displacement responses.

In the two-dimensional setting, the regularization strategy was therefore modified to a log-barrier-type formulation with the objective of fully suppressing these oscillations. For the parameter values considered in the present study, a saw-tooth response was still observed, indicating that the oscillations are reduced but not completely eliminated. It is nevertheless plausible that alternative choices of the barrier parameter may further smooth the global response, a possibility that was not explored here.

The use of inequality- or barrier-based regularization in the present work is motivated by the desire to delay the influence of strain-gradient effects, rather than activating them from the onset of deformation as in classical gradient elasticity formulations such as those of Mindlin or Toupin. In this sense, the regularization acts primarily as a stabilization mechanism that becomes relevant only once localization develops, rather than as a constitutive modification that alters the material response throughout the loading process.

Finally, the alternating minimization strategy employed in the present work bears conceptual similarities to search-direction-based methods such as LATIN [21]. Exploring hybrid formulations that combine data-driven material representations with such solution strategies represents a promising direction for future research.

## 8. Conclusion

This work examined the behavior of DDCM minimization algorithm in the presence of damage-induced softening and snapback instabilities. By analytically defining the material response, database discretization effects have been removed and the algorithmic aspects have been isolated.

It has been shown that, for softening materials, the convergence and correctness of DD solutions depend critically on the choice of hyper-parameters and on the use of continuation strategies. In particular, hyper-parameters consistent with the local material state tangent response are necessary to obtain stable convergence and to follow localized damage branches. Moreover, displacement control alone has been found to be insufficient to traverse snapback instabilities, while arc-length continuation enables robust tracking of bifurcated equilibrium paths.

In the absence of regularization, the data-driven formulation reproduces the expected mesh-dependent localization behavior, provided the algorithmic ingredients are chosen appropriately. A strain-gradient-based regularization strategy was then introduced within the DD setting, yielding mesh-objective localization widths in both one- and two-dimensional problems.

Overall, the results demonstrate that damage and fracture can be consistently addressed within the data-driven paradigm, provided that non-convexity is handled through appropriate algorithmic choices. These findings establish

essential guidelines for extending data-driven methods to failure problems and form a foundation for future studies based on experimentally derived material databases.

## A. Analysis of convergence

The mechanical and material updates in the 1D case will be described here. Restating the problem, the element in the middle is already taken to lay on the damage branch, while the other elements are unloading on the elastic branch. See Figure 1. The distance functional is

$$d(\sigma, \sigma_1^*, \sigma_2^*, \epsilon_1, \epsilon_2; u_g) = \frac{N\ell_e}{2} \left[ \frac{(\sigma - \sigma_1^*)^2}{C_1} + C_1(\epsilon_1 - \epsilon_1^*)^2 \right] + \frac{\ell_e}{2} \left[ \frac{(\sigma - \sigma_2^*)^2}{C_2} + C_2(\epsilon_2 - \epsilon_2^*)^2 \right].$$

The compatibility condition is  $N\epsilon_1 + \epsilon_2 = u_D/\ell_e$ . The mechanical state of the bar is described by a scalar stress  $\sigma \in \mathbb{R}$  and strains  $\epsilon_1, \epsilon_2 \in \mathbb{R}$  ( $\epsilon_1$  for the undamaged elements and  $\epsilon_2$  for the damaged element). The material state is described by  $(\epsilon_1^*, \sigma_1^*) \in \mathbb{R}^{1 \times 1}$  and  $(\epsilon_2^*, \sigma_2^*) \in \mathbb{R}^{1 \times 1}$ .

Now, taking the variations with respect to the compatibility equation as a constraint, the system of equations for  $\epsilon_1, \epsilon_2$  and  $\sigma$  are obtained as

$$\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \sigma \end{bmatrix} = \begin{bmatrix} C_1/(C_1 + NC_2) & -C_2/(C_1 + NC_2) & 0 & 0 \\ -NC_1/(C_1 + NC_2) & NC_2/(C_1 + NC_2) & 0 & 0 \\ 0 & 0 & \frac{NC_2}{C_1+NC_2} & \frac{C_1}{C_1+NC_2} \end{bmatrix} \begin{bmatrix} \epsilon_1^* \\ \epsilon_2^* \\ \sigma_1^* \\ \sigma_2^* \end{bmatrix} + \begin{bmatrix} C_2 u_D/(C_1 + NC_2)/\ell_e \\ C_1 u_D/(C_1 + NC_2)/\ell_e \\ 0 \end{bmatrix}. \quad (38)$$

The material states can be found by taking the variations with respect to  $\epsilon_1^*, \epsilon_2^*$ , while realizing that  $\sigma_1^* = E\epsilon_1^*$ , and  $\sigma_2^* = m\epsilon_2^* + c$ . Thus,

$$\begin{bmatrix} \epsilon_1^* \\ \epsilon_2^* \\ \sigma_1^* \\ \sigma_2^* \end{bmatrix} = \begin{bmatrix} C_1^2/(E^2 + C_1^2) & 0 & E/(E^2 + C_1^2) \\ 0 & C_2^2/(m^2 + C_2^2) & m/(m^2 + C_2^2) \\ EC_1^2/(E^2 + C_1^2) & 0 & E^2/(E^2 + C_1^2) \\ 0 & mC_2^2/(m^2 + C_2^2) & m^2/(m^2 + C_2^2) \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \sigma \end{bmatrix} + \begin{bmatrix} 0 \\ -mc/(m^2 + C_2^2) \\ 0 \\ cC_2^2/(m^2 + C_2^2) \end{bmatrix}. \quad (39)$$

The system of equations corresponding to the mechanical and material updates can be written in short as  $\mathbf{a} = \mathbf{Ab} + \mathbf{c}$ , and  $\mathbf{b} = \mathbf{Ca} + \mathbf{d}$ . The alternating minimization can then be written as

$$\begin{aligned} \mathbf{b}^{n+1} &= \mathbf{Ca}^n + \mathbf{d}, \\ \mathbf{a}^{n+1} &= \mathbf{Ab}^{n+1} + \mathbf{c}. \end{aligned}$$

Combining them together,

$$\mathbf{a}^{n+1} = \mathbf{ACa}^n + \mathbf{Ad} + \mathbf{c}. \quad (40)$$

At convergence,  $\mathbf{a}^{n+1} \approx \mathbf{a}^n \approx \mathbf{a}$ . Thus,  $\mathbf{e}^{n+1} = \mathbf{ACe}^n$ , where  $\mathbf{e}^{n+1} = \mathbf{a}^{n+1} - \mathbf{a}$  is the error at the iteration  $n+1$ . The rate of convergence of the method depends on the spectral radius of the matrix  $\mathbf{AC}$ , which in turn depends on the metric chosen.

### With arc length solver

Snapback occurs when the mesh and thus, the size of the damage zone, is refined. In this case, an arc length solver is needed to track the full damage evolution in the element. So, this case will first be presented and the effect of metric on the rate of convergence will be studied.

Using an arc length solver corresponds to when the displacement at the end of the bar,  $u_D$ , is treated as an unknown. This results in the number of unknowns increasing by 1. The additional equation that is needed to fully resolve the system is the arc length constraint, stated in equation 56 is  $\epsilon_2 = \epsilon_0 + \Delta\epsilon$ . This changes the equations to be resolved in the previous section to

$$\begin{bmatrix} 1 & 0 & 0 & -C_2/(C_1 + NC_2)\ell_e \\ 0 & 1 & 0 & -C_1/(C_1 + NC_2)\ell_e \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \sigma \\ u_D \end{bmatrix} = \begin{bmatrix} C_1/(C_1 + NC_2) & -C_2/(C_1 + NC_2) & 0 & 0 \\ -NC_1/(C_1 + NC_2) & NC_2/(C_1 + NC_2) & 0 & 0 \\ 0 & 0 & \frac{NC_2}{C_1+NC_2} & \frac{C_1}{C_1+NC_2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1^* \\ \epsilon_2^* \\ \sigma_1^* \\ \sigma_2^* \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \epsilon_0 + \Delta\epsilon \end{bmatrix}. \quad (41)$$

The material updates remain unchanged and can be found by taking the variations with respect to  $\epsilon_1^*$ ,  $\epsilon_2^*$ , while realizing that  $\sigma_1^* = E\epsilon_1^*$ , and  $\sigma_2^* = m\epsilon_2^* + c$ . Thus,

$$\begin{bmatrix} \epsilon_1^* \\ \epsilon_2^* \\ \sigma_1^* \\ \sigma_2^* \end{bmatrix} = \begin{bmatrix} C_1^2/(E^2 + C_1^2) & 0 & E/(E^2 + C_1^2) & 0 \\ 0 & C_2^2/(m^2 + C_2^2) & m/(m^2 + C_2^2) & 0 \\ EC_1^2/(E^2 + C_1^2) & 0 & E^2/(E^2 + C_1^2) & 0 \\ 0 & mC_2^2/(m^2 + C_2^2) & m^2/(m^2 + C_2^2) & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \sigma \\ u_D \end{bmatrix} + \begin{bmatrix} 0 \\ -mc/(m^2 + C_2^2) \\ 0 \\ cC_2^2/(m^2 + C_2^2) \end{bmatrix}. \quad (42)$$

Thus, even in this case, the update can be cast into the form of equation 40 and can be analyzed similarly.

## B. Discrete version of the constraint

To impose the constraint on the strain variable, a dual mesh (also called the lip mesh) is defined along with the FE mesh as in [7]. To compute the derivative of the strain variable, in discrete formulation, the lip mesh is used. The strain variables are computed from the displacements as  $\boldsymbol{\epsilon} = \mathbf{B}\mathbf{u}$ , where  $\mathbf{B}$  is the discrete gradient operator, computed on the FE mesh. Defining the discrete gradient operator on the lip mesh and representing it in a matrix form as  $\mathbf{G}$ , the gradients of strain can be expressed in terms of the displacements as

$$\mathbf{k} = \nabla\boldsymbol{\epsilon} = \mathbf{B}'\mathbf{B}\mathbf{u} = \mathbf{G}\mathbf{u}. \quad (43)$$

Where,  $\mathbf{k}$  is the matrix containing the gradients of strain as  $[k_{111} \ k_{211} \ k_{122} \ k_{222} \ k_{112} \ k_{212}]^T$ .

The constraint on the strain gradient can thus be written as

$$(\mathbf{G}\mathbf{u})^T \mathbf{G}\mathbf{u} \leq \frac{1}{\ell_c^2}, \quad (44)$$

Figure 23 compares the energy responses of different terms used to enforce the strain-gradient constraint. The standard quadratic form increases rapidly for small deviations from zero, leading to early activation near the origin. In contrast, the log-barrier-type term remains nearly flat in a neighborhood of zero, thereby postponing the onset of localization until the strain gradients begin to concentrate.

## C. Material update

When the mechanical state is given, the material state can be updated analytically in the present case.

### In 1D

In 1D, assuming that the database is represented by a model of the form  $\sigma^* = m\epsilon^* + c$ , the material state is the minimizer of the following functional:

$$\frac{(\sigma - \sigma^*)^2}{2C} + \frac{C}{2}(\epsilon - \epsilon^*)^2. \quad (45)$$

Taking its derivative with respect to  $\epsilon^*$  and equating it to zero, the material state can be seen to be

$$\epsilon^* = \frac{m\sigma + C^2\epsilon - mc}{m^2 + C^2}, \quad \sigma^* = m\epsilon^* + c. \quad (46)$$

The point on the elastic branch can be found by simply using  $m = E, c = 0$  in the above equation.

$$\epsilon^* = \frac{E\sigma + C^2\epsilon}{E^2 + C^2}, \quad \sigma^* = E\epsilon^*. \quad (47)$$

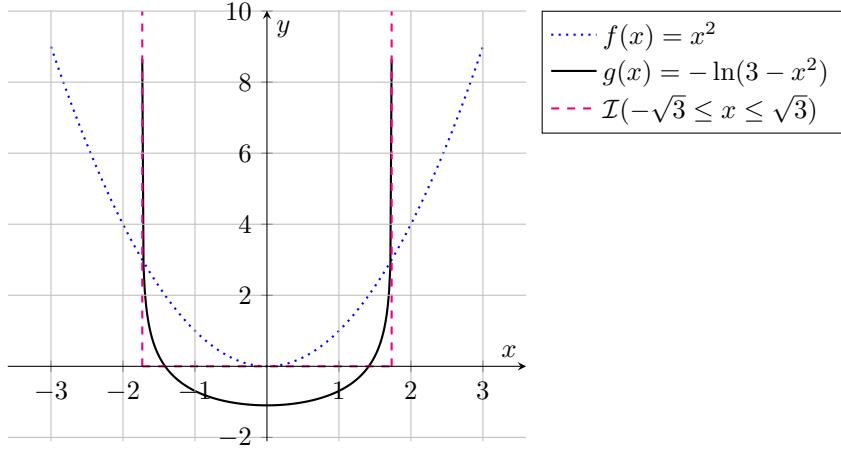


Figure 23: Log barrier function vs quadratic functional

## In 2D

The generating model is  $\sigma^*_i = g(D)E_{ij}\epsilon^*_j$ .  $g$  is the degradation function of the damage variable,  $D$ . The damage evolution in the present case is taken to be a function of strain as

$$D = \frac{\sqrt{k\psi_0(\epsilon^*)} - \sqrt{Y_c}}{(k-1)\sqrt{Y_c}}. \quad (48)$$

$\psi_0$  is the strain energy density.  $Y_c$  and  $k$  are material parameters.  $g$  is then defined as

$$g(D) = \frac{1-D}{1+(k-1)D}. \quad (49)$$

The distance functional, written in index form is

$$d(\dots) = \frac{1}{2}(\sigma - \sigma^*)_i S_{ij}(\sigma - \sigma^*)_j + \frac{1}{2}(\epsilon - \epsilon^*)_i C_{ij}(\epsilon - \epsilon^*)_j. \quad (50)$$

The material state can then be found by minimizing the above functional with respect to  $\epsilon^*$ . Thus,  $\frac{\partial d}{\partial \epsilon^*} = 0$ . This can be further developed as follows:

$$\frac{\partial d}{\partial \epsilon^*_k} = -\frac{\partial \sigma^*_i}{\partial \epsilon^*_k} S_{ij}(\sigma - \sigma^*)_j - \frac{\partial \epsilon^*_i}{\partial \epsilon^*_k} C_{ij}(\epsilon - \epsilon^*)_j. \quad (51)$$

Realizing that  $\sigma^*_i = g(d)E_{ij}\epsilon^*_j$ , the derivative can be written as

$$\frac{\partial \sigma^*_i}{\partial \epsilon^*_k} = g'(D) \frac{\partial D}{\partial \epsilon^*_k} E_{ij}\epsilon^*_j + g(D)E_{ij}\delta_{jk}, \quad (52)$$

$$= -\frac{\sqrt{kY_c}}{2(k-1)\psi_0\sqrt{\psi_0}} \hat{\sigma}^*_i \hat{\sigma}^*_k + g(D)E_{ik}. \quad (53)$$

Thus, the equation to be resolved can be seen to be

$$r_k = \left( \frac{\sqrt{kY_c}}{2(k-1)\psi_0\sqrt{\psi_0}} \hat{\sigma}^*_i \hat{\sigma}^*_k + g(D)E_{ik} \right) S_{ij}(\sigma - \sigma^*)_j - C_{kj}(\epsilon - \epsilon^*)_j = 0. \quad (54)$$

The above equation is non-linear and requires newton's method to solve. This needs the derivative of residual, which can be seen to be

$$\frac{\partial r_k}{\partial \epsilon^*_p} = \left\{ \frac{\sqrt{kY_c}}{2(k-1)} \left[ \frac{E_{kp}\hat{\sigma}^*_i + \hat{\sigma}^*_k E_{ip} + \hat{\sigma}^*_p E_{ik}}{\psi_0 \sqrt{\psi_0}} - \frac{3}{2} \psi_0^{-5/2} \hat{\sigma}^*_i \hat{\sigma}^*_k \hat{\sigma}^*_p \right] S_{ij} (\sigma - \sigma^*)_j \right\} + \\ \left( \frac{\sqrt{kY_c}}{2(k-1)\psi_0 \sqrt{\psi_0}} \hat{\sigma}^*_i \hat{\sigma}^*_k + g(D) E_{ik} \right) S_{ij} \left( \frac{\sqrt{kY_c}}{2(k-1)\psi_0 \sqrt{\psi_0}} \hat{\sigma}^*_j \hat{\sigma}^*_p + g(D) E_{jp} \right) + C_{kp}. \quad (55)$$

The elastic case can be obtained by taking  $g' = 0$  in equation 52.

## D. Arc length constraint

The arc length constraint is simpler in 1D case than in 2D case as it involves selecting degrees of freedom where a certain quantity is monotone increasing. For instance, in [], the damage variables are used to write the arc length constraint. In the current case, for the 1D bar, the mechanical strain variable at the left end of the bar is chosen to impose the arc length constraint. The constraint can be expressed as

$$\epsilon^{n+1} - \epsilon^n = \Delta, \quad (56)$$

where  $n + 1$  and  $n$  denote the load steps.  $\Delta$  is the arc length, which in the current case is the strain increment for the element at the extreme left between two steps. Expressing the displacements as  $\mathbf{u} = \mathbf{C}_f \mathbf{u}_f + \lambda \mathbf{u}_g$ , where  $\mathbf{u}_f$  denotes the unknown/free degrees of freedom and  $\lambda$ , in the current case, is the displacement applied at the right end with  $\mathbf{u}_g$  as the corresponding degree of freedom. For instance, if the bar contains  $N$  nodes numbered 1 to  $N$ ,  $\mathbf{u}_f$  contains the DOFs of the nodes 2 to  $N - 1$ .  $\mathbf{C}_f \in \mathbb{R}^{N \times N-2}$  and  $\mathbf{u}_g \in \mathbb{R}^N = \{0, 0, 0, \dots, 1\}^T$ .

In 2D case, the method is slightly involved. The aim is to identify the degrees of freedom where a specific quantity is always increasing. For this purpose, the elements whose material states lie on the 'damage branch' are first identified. The damage levels in these elements are thus increasing in the load step under consideration. Then, an equivalent strain energy density of the mechanical states corresponding to these elements are computed using the mechanical strains. The arc length constraint is now written using these degrees of freedom and their equivalent strain energy density. Let the damage-driving energy over an element  $e$  be

$$\psi_0^{(e)} = \frac{1}{2} \boldsymbol{\epsilon}^{(e)T} \mathbf{C} \boldsymbol{\epsilon}^{(e)}. \quad (57)$$

The damage evolution term is defined as

$$d^{(e)} = \frac{\sqrt{k \psi_0^{(e)}} - \sqrt{Y_c}}{\sqrt{Y_c}(k-1)}. \quad (58)$$

The arc-length constraint is now written as

$$r_{diss} = \sum_e A_e \left[ \frac{\sqrt{k \psi_0^{(e)}} - \sqrt{Y_c}}{\sqrt{Y_c}(k-1)} - d_{old}^{(e)} \right] - \Delta = 0. \quad (59)$$

This equation is added to the system of equations 32 - 37 and solved using the Newton method.

## References

- [1] Andrés A. León Baldelli and Corrado Maurini. "Numerical bifurcation and stability analysis of variational gradient-damage models for phase-field fracture". In: *Journal of the Mechanics and Physics of Solids* 152 (July 2021), p. 104424. ISSN: 0022-5096. DOI: 10.1016/J.JMPS.2021.104424.
- [2] Ritukesh Bharali et al. "A robust monolithic solver for phase-field fracture integrated with fracture energy based arc-length method and under-relaxation". In: *Computer Methods in Applied Mechanics and Engineering* 394 (2022), p. 114927.

- [3] R. de Borst. "Computation of post-bifurcation and post-failure behavior of strain-softening solids". In: *Computers and Structures* 25 (2 Jan. 1987), pp. 211–224. ISSN: 0045-7949. DOI: 10.1016/0045-7949(87)90144-1.
- [4] Rene De Borst. "Non-linear analysis of frictional materials". PhD thesis. 1986. URL: <https://repository.tudelft.nl/record/uuid:d52ac7f8-d56f-4e0c-ab14-455ed02fcffe>.
- [5] P. Carrara, M. Ortiz, and L. De Lorenzis. "Data-driven rate-dependent fracture mechanics". In: *Journal of the Mechanics and Physics of Solids* 155 (Oct. 2021). ISSN: 00225096. DOI: 10.1016/j.jmps.2021.104559.
- [6] P. Carrara et al. "Data-driven fracture mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 372 (Dec. 2020). ISSN: 00457825. DOI: 10.1016/j.cma.2020.113390.
- [7] N. Chevaugeon and N. Moes. "Lipschitz regularization for fracture: the Lip-field approach". In: *Computer Methods in Applied Mechanics and Engineering* (xxxx 2021), p. 115644. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.115644. URL: <http://arxiv.org/abs/2111.04771>.
- [8] S. Conti, S. Müller, and M. Ortiz. "Data-Driven Problems in Elasticity". In: *Archive for Rational Mechanics and Analysis* 229 (1 2018), pp. 79–123. ISSN: 14320673. DOI: 10.1007/s00205-017-1214-0. URL: <https://doi.org/10.1007/s00205-017-1214-0>.
- [9] Héloïse Dandin, Adrien Leygue, and Laurent Stainier. "Graph-based representation of history-dependent material response in the Data-Driven Computational Mechanics framework". In: *Computer Methods in Applied Mechanics and Engineering* 419 (2024), p. 116694. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2023.116694>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782523008174>.
- [10] R. Eggersmann et al. "Model-Free Data-Driven inelasticity". In: *Computer Methods in Applied Mechanics and Engineering* 350 (2019), pp. 81–99. ISSN: 00457825. DOI: 10.1016/j.cma.2019.02.016. URL: <https://doi.org/10.1016/j.cma.2019.02.016>.
- [11] Robert Eggersmann et al. "Efficient data structures for model-free data-driven computational mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 382 (2021), p. 113855. ISSN: 00457825. DOI: 10.1016/j.cma.2021.113855. URL: <https://doi.org/10.1016/j.cma.2021.113855>.
- [12] Robert Eggersmann et al. "Model-free data-driven computational mechanics enhanced by tensor voting". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113499. ISSN: 00457825. DOI: 10.1016/j.cma.2020.113499. URL: <https://doi.org/10.1016/j.cma.2020.113499>.
- [13] Patrick Farrell and Corrado Maurini. "Linear and nonlinear solvers for variational phase-field models of brittle fracture". In: *International Journal for Numerical Methods in Engineering* 109 (5 Feb. 2017), pp. 648–667. ISSN: 1097-0207. DOI: 10.1002/NME.5300. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/nme.5300> <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5300>.
- [14] M. G.D. Geers et al. "Strain-based transient-gradient damage model for failure analyses". In: *Computer Methods in Applied Mechanics and Engineering* 160 (1-2 July 1998), pp. 133–153. ISSN: 0045-7825. DOI: 10.1016/S0045-7825(98)80011-X.
- [15] T. Gerasimov and L. De Lorenzis. "A line search assisted monolithic approach for phase-field computing of brittle fracture". In: *Computer Methods in Applied Mechanics and Engineering* 312 (Dec. 2016), pp. 276–303. ISSN: 0045-7825. DOI: 10.1016/J.CMA.2015.12.017.
- [16] Vasudevan Kamasamudram and Laurent Stainier. "A strain based Lipschitz regularization for materials undergoing damage". In: *Comptes Rendus - Mecanique* 351 (G1 2023), pp. 125–149. ISSN: 18737234. DOI: 10.5802/CRMeca.176/. URL: <http://creativecommons.org/licenses/by/4.0/> LesComptesRendus. M&A;caniquesontmembresduCentreMersennepour l'Ã‰ditionscientifiqueouverte [www.centre-mersenne.org](http://www.centre-mersenne.org) <https://doi.org/10.5802/crmeca.176>.
- [17] Yoshihiro Kanno. "Mixed-integer programming formulation of a data-driven solver in computational elasticity". In: *Optimization Letters* 13 (7 2019), pp. 1505–1514. ISSN: 18624480. DOI: 10.1007/s11590-019-01409-w. URL: <https://doi.org/10.1007/s11590-019-01409-w>.
- [18] K. Karapiperis, M. Ortiz, and J. E. Andrade. "Data-Driven nonlocal mechanics: Discovering the internal length scales of materials". In: *Computer Methods in Applied Mechanics and Engineering* 386 (2021), p. 114039. ISSN: 00457825. DOI: 10.1016/j.cma.2021.114039. URL: <https://doi.org/10.1016/j.cma.2021.114039>.
- [19] T. Kirchdoerfer and M. Ortiz. "Data-driven computational mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 304 (2016), pp. 81–101. ISSN: 00457825. DOI: 10.1016/j.cma.2016.02.001. URL: <http://dx.doi.org/10.1016/j.cma.2016.02.001>.
- [20] Zengtao Kuang et al. "Data-driven computational framework for snap-through problems". In: *International Journal of Solids and Structures* 269 (May 2023), p. 112226. ISSN: 0020-7683. DOI: 10.1016/J.IJSOLSTR.2023.112226.
- [21] Pierre Ladevèze. "The large time increment method for the analysis of structures with non-linear behavior described by internal variables". In: *COMPTEES RENDUS DE L ACADEMIE DES SCIENCES SERIE II* 309.11 (1989), pp. 1095–1099.
- [22] Duc Trung Le et al. "Strain-gradient vs damage-gradient regularizations of softening damage models". In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 424–450. ISSN: 00457825. DOI: 10.1016/j.cma.2018.06.013. URL: <https://doi.org/10.1016/j.cma.2018.06.013>.
- [23] Duc Trung Le et al. "Strain-gradient vs damage-gradient regularizations of softening damage models". In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 424–450.
- [24] Raymond David Mindlin and NN0166 Eshel. "On first strain-gradient theories in linear elasticity". In: *International Journal of Solids and Structures* 4 (1 1968), pp. 109–124. ISSN: 0020-7683.

- [25] Lu Trong Khiem Nguyen, Roland Can Aydin, and Christian Johannes Cyron. "Accelerating the distance-minimizing method for data-driven elasticity with adaptive hyperparameters". In: *Computational Mechanics* 70 (3 Sept. 2022), pp. 621–638. ISSN: 14320924. DOI: 10.1007/S00466-022-02183-w. URL: <https://link.springer.com/article/10.1007/s00466-022-02183-w>.
- [26] Thi Hai Nhu Nguyen. "Contributions to multiscale modelling of quasi-brittle damage in heterogeneous materials". PhD thesis. 2019.
- [27] Vinh Phu Nguyen, Martijn Stroeven, and Lambertus Johannes Sluys. "An enhanced continuous-discontinuous multiscale method for modeling mode-I cohesive failure in random heterogeneous quasi-brittle materials". In: *Engineering Fracture Mechanics* 79 (2012), pp. 78–102.
- [28] Vinh Phu Nguyen et al. "On the existence of representative volumes for softening quasi-brittle materials—a failure zone averaging scheme". In: *Computer Methods in Applied Mechanics and Engineering* 199.45–48 (2010), pp. 3028–3038.
- [29] Duc Khai Nguyen Pham, Nawfal Blal, and Anthony Gravouil. "Tangent space Data Driven framework for elasto-plastic material behaviors". In: *Finite Elements in Analysis and Design* 216 (Apr. 2023), p. 103895. ISSN: 0168-874X. DOI: 10.1016/J.FINEL.2022.103895.
- [30] Gilles Pijaudier-Cabot et al. "A review of non local continuum damage: Modelling of failure?" In: *Networks and Heterogeneous Media* 9 (4 2014), pp. 575–597. ISSN: 1556-1801. DOI: 10.3934/NHM.2014.9.575. URL: [https://www.aims sciences.org/en/article/doi/10.3934/nhm.2014.9.575?viewType=HTML](https://www.aims sciences.org/en/article/doi/10.3934/nhm.2014.9.575) <https://www.aims sciences.org/article/doi/10.3934/nhm.2014.9.575>.
- [31] Felipe Rocha et al. "On-the-fly adaptive sampling strategy for data-driven computational mechanics: Applications to computational homogenisation". In: *Mechanics of Materials* 207 (2025), p. 105382. ISSN: 0167-6636. DOI: <https://doi.org/10.1016/j.mechmat.2025.105382>. URL: <https://www.sciencedirect.com/science/article/pii/S0167663625001449>.
- [32] N. Singh et al. "A fracture-controlled path-following technique for phase-field modeling of brittle fracture". In: *Finite Elements in Analysis and Design* 113 (2016), pp. 14–29. ISSN: 0168874X. DOI: 10.1016/j.finel.2015.12.005. URL: <http://dx.doi.org/10.1016/j.finel.2015.12.005>.
- [33] Jacinto Ulloa et al. "Data-driven micromorphic mechanics for materials with strain localization". In: *Computer Methods in Applied Mechanics and Engineering* 429 (2024), p. 117180. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2024.117180>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782524004365>.
- [34] Z. Cedric Xia and John W. Hutchinson. "Crack tip fields in strain gradient plasticity". In: *Journal of the Mechanics and Physics of Solids* 44 (10 1996), pp. 1621–1648. ISSN: 00225096. DOI: 10.1016/0022-5096(96)00035-X.