



HAL
open science

Concurrent development of local and non-local damage with the Thick Level Set approach: Implementation aspects and application to quasi-brittle failure

Kevin Moreau, Nicolas Moes, Alexis Salzman, Nicolas Chevaugeon

► To cite this version:

Kevin Moreau, Nicolas Moes, Alexis Salzman, Nicolas Chevaugeon. Concurrent development of local and non-local damage with the Thick Level Set approach: Implementation aspects and application to quasi-brittle failure. *Computer Methods in Applied Mechanics and Engineering*, 2017, 327, pp.306-326. 10.1016/j.cma.2017.08.045 . hal-03928065

HAL Id: hal-03928065

<https://hal.science/hal-03928065>

Submitted on 6 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Concurrent development of local and non-local damage with the Thick Level Set approach: Implementation aspects and application to quasi-brittle failure

Kévin Moreau, Nicolas Moës*, Nicolas Chevaugeon, Alexis Salzman

Ecole Centrale de Nantes, GeM Institute, UMR CNRS 6183, 1 rue de la Noe, 44321 Nantes, France

The present paper focuses on the discretization and implementation of the Thick Level Set (TLS) approach for quasi-brittle failure when dealing with both local and non-local development of damage.

The TLS damage model introduces a length scale and nonlocality in the Fracture Process Zone (FPZ) and it handles the continuous to discontinuous failure transition. The added nonlocality is limited to the FPZ and therefore the model is local away from the FPZ or before its emergence. It leads to concurrent development of local and nonlocal damage during simulations. This paper presents a space–time discretization that handles concurrent computations in a unified manner. It is based on an explicit time discretization written as a predictor–corrector scheme and a space discretization that uses specific approximation functions (modes) that embed nonlocality. It also uses an alternative implementation of the eXtended Finite Element Method (X-FEM) enrichment used in the TLS which is based on virtual nodes.

Keywords: Damage; Fracture; TLS; Thick Level Set; XFEM

1. Introduction

Quasi-brittle failure is commonly characterized, at the macroscopic scale, by a strain localization followed by the emergence of a crack. Strain localization appears in a privileged region called the fracture process zone. At a lower scale, strain localization results from a microcracking process and macrocracks are the consequence of microcracks growth and coalescence. In the present paper a quasi-brittle failure model is developed at the macroscopic scale.

Modeling quasi-brittle failure can be achieved in several different ways. The first one consists in extending the linear elastic fracture mechanics by adding cohesive zones on crack tips in which cohesive tractions gradually decay as the crack faces separate [1–3]. Another way consists in using continuum damage mechanics [4–6]. Within this theory, material behavior softens through the use of a damage internal variable which models microcracking effects. Unfortunately, local rate-independent damage models are known to allow solutions which are not acceptable

* Corresponding author.

E-mail address: nicolas.moes@ec-nantes.fr (N. Moës).

(crack creation without dissipation [7]). This theoretical issue affects the numerical simulations by spurious mesh dependencies.

Several models have been proposed in the literature to make sure that solutions reach a proper level of dissipation by avoiding damage to localize in an infinitely narrow band. They can be classified into different categories as integral based [8,9], differential (gradient dependent) based or rate based [10,11]. Regarding the gradient based category, one may even distinguish between the introduction of higher order kinematics [12] or damage gradient in the description of the energy [13]. The latter category involves more recent approaches as the variational approach to fracture [14] as well as the phase-field approach [15]. A recent paper [16] details the similarities and differences between the initial gradient damage approach and the phase-field approach.

Note that, as it should be, the above models do not necessarily ensure uniqueness of the solution, bifurcations are still possible and must ideally be taken into account in the simulation. Even though the above models avoid spurious localization, the transition to a crack (displacement discontinuity) is not directly addressed by the existing models. The crack placement is an extra step. Several papers are devoted to crack placement within damaged zone, see for instance [17–19].

The TLS model follows a different line of thought. It merges the capabilities of the strong discontinuity approach and of continuum based damage model. On one hand, damage is used to predict crack initiation and to handle complex crack patterns (topological changes such as crack birth coalescence and branching). On the other hand, the discontinuity capability is used to avoid the need to keep a fine computational mesh along the path of already formed cracks.

The Thick Level Set (TLS) approach was first introduced in [20] and later improved to handle both smooth and localizing damage within the same simulation in [21]. The key TLS equation is:

$$\|\nabla d\| \leq g(d) \quad (1)$$

where d is the damage variable damage and $g(d)$ a function parameter of the model. Defining a function $d = d(\phi)$, such that

$$g(d) = \frac{dd}{d\phi} \quad (2)$$

Eq. (1) may be recast in a simpler expression

$$\|\nabla\phi\| \leq 1 \quad (3)$$

where one recognizes the Eikonal equation (up to the inequality sign), the corner stone of the level set technology [22,23]. The above shows that function ϕ is a distance function (when the inequality is fulfilled). The damage profile $d = d(\phi)$ is chosen so that the material is fully damaged ($d = 1$) when $\phi = l_c$. The crack lips are thus located on the level set $\phi = l_c$. The extraction of the crack lips is robust using ϕ since the latter is a distance function. The damage variable enters thus a classical damage model but has at the same time a geometrical interpretation through the damage profile $d(\phi)$. The TLS gives a configurational twist to the local damage model.

The present paper, discusses an efficient implementation of the inequality (3). An implementation was already discussed in [21] but it was only appropriate in the 1D setting. The scheme presented here has the advantage that damage evolution is computed without solving any linear system. Cracks location (obtained as a given level set), is modeled using the X-FEM. Note that to allow an easy implementation of the TLS even within an FEM code that does not allow node enrichment, an alternative implementation of the XFEM enrichment is detailed in the paper. It is based on the virtual node concept [24–26].

The paper is organized as follows. Next section details the TLS governing equations. The third section describes the space discretization of the formulation using the X-FEM. In the fourth section, we propose an explicit time discretization based on a prediction–correction scheme. Finally, the full implementation is detailed in the fifth section. Section six is dedicated to numerical applications. A last section proposes a discussion.

2. Governing equations

We model quasi-brittle failure considering an isotropic homogeneous elastic scalar damage constitutive model, under small deformation assumptions. The considered body occupies the domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$. We assume a quasistatic evolution and introduce a time-like parameter t belonging to the time interval $T = [0, t_f]$.

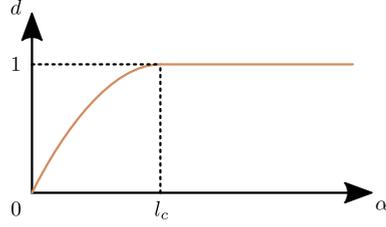


Fig. 1. Example of damage shape function.

2.1. Damage field

Within the TLS [20,21], the damage field

$$\begin{aligned} d : \overline{\Omega} \times \mathbb{T} &\rightarrow [0, 1] \\ (\mathbf{x}, t) &\mapsto d(\mathbf{x}, t), \end{aligned} \quad (4)$$

is deduced from another field, the level set field, denoted as ϕ ,

$$\begin{aligned} \phi : \overline{\Omega} \times \mathbb{T} &\rightarrow \mathbb{R}_+ \\ (\mathbf{x}, t) &\mapsto \phi(\mathbf{x}, t). \end{aligned} \quad (5)$$

This field must satisfy the following inequality

$$\|\nabla\phi\| \leq 1, \quad (6)$$

where $\|\cdot\|$ indicates the euclidean norm on \mathbb{R}^3 . The damage field is then deduced from ϕ using the damage shape function

$$\begin{aligned} d : \mathbb{R}_+ &\rightarrow [0, 1] \\ \alpha &\mapsto d(\alpha), \end{aligned} \quad (7)$$

simply by writing

$$d = d(\phi). \quad (8)$$

The damage shape function is a parameter of the model, it is assumed continuous and differentiable with the following properties

$$d(0) = 0, \quad (9)$$

$$d'(\alpha) > 0 \text{ for } \alpha \in [0, l_c[, \quad (10)$$

$$d(\alpha) = 1 \text{ for } \alpha \in [l_c, +\infty[, \quad (11)$$

where l_c is a parameter called the characteristic length. An example is illustrated in Fig. 1.

Using the ϕ field, the domain $\overline{\Omega}$ is decomposed into three parts, one for which the body is fully damaged, one for which the inequality (6) is strict and one for which the inequality reduces to an equality

$$\Omega_c = \{\mathbf{x} \in \overline{\Omega} : \phi(\mathbf{x}) \geq l_c\}, \quad (\text{closed}) \quad (12)$$

$$\Omega^- = \{\mathbf{x} \in \Omega \setminus \Omega_c : \|\nabla\phi(\mathbf{x})\| < 1\}, \quad (\text{open}) \quad (13)$$

$$\Omega^+ = (\Omega \setminus \Omega_c) \setminus \overline{\Omega}^-. \quad (\text{open}) \quad (14)$$

These parts of $\overline{\Omega}$ are called the fully damaged zone, the local zone and the nonlocal zone, respectively. The fully damage zone is defined as a closed set to take into account the case where its interior is empty and as a consequence, the fully damaged zone reduces to its boundary $\Omega_c = \partial\Omega_c$. This is the case when the crack is simply a curve in 2D and a surface in 3D (the TLS does not force a priori forces the fully damaged zone to be of zero measure this is why we take some mathematical care).

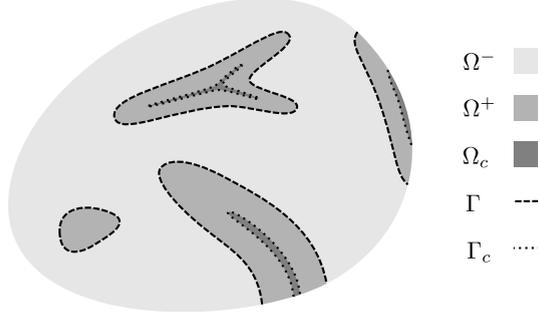


Fig. 2. Parts and interfaces involved in TLS.

The interface between the non-local zone and the local zone is denoted as Γ whereas the boundary between the non-local zone and the fully damaged zone is denoted as Γ_c .

$$\Gamma = \overline{\Omega^-} \cap \overline{\Omega^+} \quad (15)$$

$$\Gamma_c = \overline{\Omega^+} \cap \Omega_c. \quad (16)$$

The following inclusion holds true

$$\Gamma_c \subset \partial\Omega_c, \quad (17)$$

and the equality is obtained whenever $\partial\Omega_c \subset \partial\Omega^+$, which is the case either when $\dot{\Omega}_c = \emptyset$ or $\partial\Omega_c \cap \partial\Omega = \emptyset$. Fig. 2 illustrates such parts and interfaces, we can see that Γ_c models crack lips. Note that Γ_c is also

$$\Gamma_c = \text{iso-}l_c(\phi) \quad (18)$$

which is comfortable from the computational point of view. This particular point is discussed in the implementation section. We can furthermore establish the function g given in Eq. (1), using Eq. (6), by writing that, in $\Omega \setminus \Omega_c$

$$\|\nabla d(\phi)\| = d'(\phi)\|\nabla\phi\| \leq d'(\phi), \quad (19)$$

and if we introduce the reciprocal function of the damage shape function restriction to $[0, l_c[$

$$\begin{aligned} d^{-1} : [0, 1[&\rightarrow [0, l_c[\\ \beta &\mapsto d^{-1}(\beta), \end{aligned} \quad (20)$$

we finally conclude that

$$g(d) = d'(\phi(d)). \quad (21)$$

Regarding the damage evolution, we enforce the following condition on Ω^+ to maintain $\|\nabla\phi\| = 1$

$$\frac{d\|\nabla\phi\|}{dt} = \nabla\phi \cdot \nabla\dot{\phi} = 0. \quad (22)$$

The above brings nonlocality into damage evolution since by taking the material derivative of Eq. (8), we have

$$\dot{d} = d'(\phi)\dot{\phi}. \quad (23)$$

Since $\dot{\phi}$ is constant along gradient of ϕ in Ω^+ , $\dot{d}/d'(\phi)$ must also be constant along gradient of ϕ in Ω^+ . We denote the space of admissible $\dot{\phi}$ by \mathcal{A}

$$\mathcal{A} = \{a \text{ regular enough over } \Omega : \nabla\phi \cdot \nabla a = 0 \text{ in } \Omega^+\}. \quad (24)$$

Given a local field X , the associated nonlocal field \bar{X} is given by solving the following variational formulation: find $\bar{X} \in \mathcal{A}$ such that

$$\int_{\Omega \setminus \Omega_c} \overline{X} d'(\phi) a \, d\omega = \int_{\Omega \setminus \Omega_c} X d'(\phi) a \, d\omega, \quad \forall a \in \mathcal{A}. \quad (25)$$

We can observe that $\overline{X} = X$ in Ω^- . A unidimensional interpretation of (25) may be found in [21].

2.2. Mechanical problem

We assume that loading conditions on the body are prescribed on the boundary $\partial(\Omega \setminus \Omega_c)$ which is partitioned into three parts

$$\Gamma_T \cup \Gamma_u \cup \Gamma_c = \partial(\Omega \setminus \Omega_c), \quad (26)$$

$$\Gamma_T \cap \Gamma_u = \Gamma_T \cap \Gamma_c = \Gamma_u \cap \Gamma_c = \emptyset. \quad (27)$$

Note that since Ω_c evolves as damage grows, parts Γ_T and Γ_u are time dependent. Fig. 2 illustrates such a case. We introduce a time-dependent load factor λ (radial type loading for simplicity), a constant stress vector field $\mathbf{T}_d^{\text{ref}}$ defined on Γ_T and a constant displacement field $\mathbf{u}_d^{\text{ref}}$ defined on Γ_u so that prescribed boundary conditions are written $\lambda \mathbf{T}_d^{\text{ref}}$ on Γ_T and $\lambda \mathbf{u}_d^{\text{ref}}$ on Γ_u . The damage model is given by the following strong formulation

Problem 1 (Mechanical Problem). Given $\mathbf{u}_d^{\text{ref}}$ and $\mathbf{T}_d^{\text{ref}}$, find \mathbf{u} , ϕ and λ such that in $\Omega \setminus \Omega_c \times [0, t_f]$

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^\top], \quad (28)$$

$$\boldsymbol{\sigma}(\mathbf{u}, \phi) = (1 - d(\phi)) \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}), \quad (29)$$

$$\text{div } \boldsymbol{\sigma}(\mathbf{u}, \phi) = \boldsymbol{\theta}, \quad (30)$$

$$Y(\mathbf{u}) = \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}), \quad (31)$$

$$f(\mathbf{u}, \phi) = \overline{Y(\mathbf{u})} - \overline{Y_c h(d(\phi))}, \quad (32)$$

$$f(\mathbf{u}, \phi) \leq 0, \quad \dot{\phi} \geq 0, \quad f(\mathbf{u}, \phi) \dot{\phi} = 0, \quad (33)$$

$$\max_{\Omega \setminus \Omega_c} \dot{\phi} = \dot{\phi}_{\max}, \quad (34)$$

with boundary conditions

$$\mathbf{u} = \lambda \mathbf{u}_d^{\text{ref}} \text{ on } \Gamma_u, \quad (35)$$

$$\boldsymbol{\sigma}(\mathbf{u}, \phi) \cdot \mathbf{n} = \lambda \mathbf{T}_d^{\text{ref}} \text{ on } \Gamma_T, \quad (36)$$

$$\boldsymbol{\sigma}(\mathbf{u}, \phi) \cdot \mathbf{n} = \boldsymbol{\theta} \text{ on } \Gamma_c, \quad (37)$$

and initial condition at $t = 0$

$$\phi = 0 \text{ in } \overline{\Omega}. \quad (38)$$

Note that in the above, the domains Ω_c , Γ_c , Γ_u and Γ_T may evolve with time. Also, the load factor is part of the unknown and determined by (34) (dissipation control algorithm).

Regarding the notations, $\boldsymbol{\varepsilon}(\mathbf{u})$ denotes the infinitesimal strain (second order) tensor field, $\boldsymbol{\sigma}(\mathbf{u}, \phi)$ the Cauchy stress (second order) tensor field and $Y(\mathbf{u})$ the energy release rate scalar field. Function f is the threshold function written in terms of nonlocal fields and \mathbf{n} the outward normal to $\partial(\Omega \setminus \Omega_c)$. Concerning material parameters, \mathbf{C} is the elasticity tensor (fourth order), Y_c is the critical energy release rate and $h : [0, 1] \rightarrow \mathbb{R}_+$ is a strictly increasing continuous hardening function with $h(0) = 1$. Parameter $\dot{\phi}_{\max} \in \mathbb{R}_+^*$ is an algorithm control parameter imposing a maximum rate to ϕ .

3. Space discretization

The domain Ω occupied by the body is discretized with a finite element mesh. The approximated domain is denoted as Ω^h . To each node i belonging to the set of nodes S , we associate a hat function N_i which is piecewise linear on

each element connected to node i . More generally, we denote by L_i any approximation function. It can be associated to a node, an edge or an element.

The first section gives approximations of displacement field and level set field. Then, the second section presents approximations of any nonlocal field and field $\dot{\phi}$. These approximations are particular to the TLS approach and are based on modes.

The notion of modes has already been used in the TLS approach [27] and modes were computed using a streamline Upwind Petrov–Galerkin formulation. Current computation of modes is rather based on the Fast-Marching Method. This is discussed in the implementation section. A third and last section applies the space discretization of Problem 1.

3.1. Approximation of the displacement and level set fields

We approximate the displacement field \mathbf{u} by

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i \in \mathbf{U}} u_i \mathbf{L}_i(\mathbf{x}) = \sum_{i \in \mathbf{R}} \sum_{j=1}^3 u_{\varphi(i,j)} \mathbf{L}_i(\mathbf{x}) \mathbf{e}_j \quad (39)$$

where

- $(u_i)_{i \in \mathbf{U}}$ are scalar approximation coefficients (the fact that the coefficient are scalars and not vectorial will allow to write easily the stiffness matrix),
- \mathbf{U} is the set of approximation coefficient ids,
- \mathbf{R} is the set of node ids in the mesh,
- $(\mathbf{e}_i)_{i \in \{1,2,3\}}$ are natural basis vectors of \mathbb{R}^3 ,
- $\varphi : \mathbf{R} \times \{1, 2, 3\} \rightarrow \mathbf{U}$ is a one-to-one and onto map,
- $\mathbf{L}_{\varphi(i,j)}(\mathbf{x}) = L_i(\mathbf{x}) \mathbf{e}_j$ with $(i, j) \in \mathbf{R} \times \{1, 2, 3\}$.

To take into account prescribed displacements, we introduce the following sets

$$\mathbf{U}_{\text{fixed}} = \{\varphi(i, j) : (i, j) \in \mathbf{R} \times \{1, 2, 3\} \text{ and } \mathbf{x}_i \in \Gamma_u^h\}, \quad \mathbf{U}_{\text{dof}} = \mathbf{U} \setminus \mathbf{U}_{\text{fixed}}, \quad (40)$$

where \mathbf{x}_i is the location of node i . We denote $(u_d^{\text{ref}})_{\varphi(i,j)} = \mathbf{u}_d^{\text{ref}}(\mathbf{x}_i) \cdot \mathbf{e}_j$ with $(i, j) \in \mathbf{R} \times \{1, 2, 3\}$ and $\varphi(i, j) \in \mathbf{U}_{\text{fixed}}$. Note that more general boundary conditions can be prescribed (component-by-component) but this one agrees with Eq. (35).

We approximate the ϕ field as

$$\phi^h(\mathbf{x}) = \sum_{i \in \mathbf{S}} \phi_i N_i(\mathbf{x}), \quad (41)$$

where ϕ_i are degrees of freedom. Gradient field is then given by

$$\nabla \phi^h(\mathbf{x}) = \sum_{i \in \mathbf{S}} \phi_i \nabla N_i(\mathbf{x}). \quad (42)$$

As further described in the implementation section, both ϕ and \mathbf{u} fields are enriched. This enrichment is realized by adding virtual nodes to the mesh and revising mesh connectivities. A consequence of enriching the approximation in such a way is that Eqs. (39) and (41) remain valid, only sets \mathbf{S} , \mathbf{R} and related change.

3.2. Nonlocal fields and $\dot{\phi}$ approximations

We first realize a partition of the set of node numbers \mathbf{S} into a local set \mathbf{S}_L and a nonlocal set \mathbf{S}_{NL} . The nonlocal set is built by looking for finite elements such that

$$\|\nabla \phi^h(\mathbf{x})\| \geq 1, \quad (43)$$

where \mathbf{x} is here the location of a point inside the considered finite element. Eq. (42) is used to evaluate the gradient. All nodes of such finite elements are classified in \mathbf{S}_{NL} . We obtain the local set as $\mathbf{S}_L = \mathbf{S} \setminus \mathbf{S}_{NL}$. We further realize a partition of \mathbf{S}_{NL} into a nonlocal boundary set \mathbf{S}_{NLB} and a nonlocal interior set \mathbf{S}_{NLI} by studying nodes adjacencies.

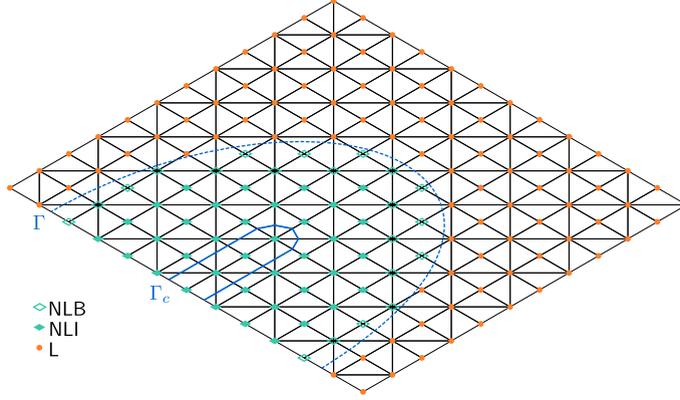


Fig. 3. Nodal classification.

For each node classified in S_{NL} if at least one adjacent node is classified in S_L then we classify the node in S_{NLB} . We obtain the nonlocal interior set as $S_{NLI} = S_{NL} \setminus S_{NLB}$. To summarize, we have partitioned the nodes in the mesh into a local and a non-local subset, (44). The non-local nodes are then partitioned into the boundary and interior nodes, (45). Finally the set of modes S_M is linked to the local and boundary non-local nodes, (46). Fig. 3 illustrates the partitions.

$$S = S_L \cup S_{NL}, \quad S_L \cap S_{NL} = \emptyset \quad (44)$$

$$S_{NL} = S_{NLB} \cup S_{NLI}, \quad S_{NLB} \cap S_{NLI} = \emptyset \quad (45)$$

$$S_M = S_L \cup S_{NLB}. \quad (46)$$

As in [27], modes are introduced to discretize the space \mathcal{A} , we write

$$\mathcal{A}^h = \left\{ a^h(\mathbf{x}) = \sum_{i \in S_M} a_i M_i(\mathbf{x}), \quad a_i \in \mathbb{R}_+ \right\} \quad (47)$$

where modes are written as a linear combination of piecewise linear finite element shape functions. It reads

$$M_i(\mathbf{x}) = \sum_{j \in S} M_{ij} N_j(\mathbf{x}), \quad \forall i \in S_M, \quad (48)$$

where $M_{ij} \in \mathbb{R}_+$ are computed in order to verify

$$M_i(\mathbf{x}) = N_i(\mathbf{x}), \quad \forall i \in S_L, \quad (49)$$

$$\nabla \phi^h \cdot \nabla M_i = 0, \quad \forall i \in S_{NLB}. \quad (50)$$

Once modes are created, any field belonging to function space \mathcal{A} can be approximated as an element of \mathcal{A}^h . We approximate any nonlocal field \bar{X} using modes

$$\bar{X}^h(\mathbf{x}) = \sum_{i \in S_M} \bar{X}_i M_i(\mathbf{x}), \quad (51)$$

where \bar{X}_i are coefficients computed from the local field X with

$$\bar{X}_i = \frac{\int_{\Omega^h \setminus \Omega_c^h} X d'(\phi^h) M_i d\omega}{\int_{\Omega^h \setminus \Omega_c^h} d'(\phi^h) M_i d\omega}, \quad \forall i \in S_M. \quad (52)$$

In the following, we write Eq. (52) in a compact manner with

$$\bar{X}_i = \langle X, d'(\phi^h) M_i \rangle, \quad \forall i \in S_M. \quad (53)$$

To obtain Eq. (52), we injected approximation (51) of \bar{X} and a in Eq. (25) and we lumped the involved matrix. In practice, Eq. (52) is not computed directly but is rather obtained from basic algebra and finite element assembling procedure, this is further described in the implementation section. We approximate $\dot{\phi}$ in the same way

$$\dot{\phi}^h(\mathbf{x}) = \sum_{i \in \text{SM}} \dot{\phi}_i M_i(\mathbf{x}). \quad (54)$$

3.3. Mechanical problem discretization

All of the material needed to discretize Problem 1 in space is given. It reads

Problem 2 (Space Discretized Mechanical Problem). At any time t , given $\mathbf{u}_d^{\text{ref},h}$ and $\mathbf{T}_d^{\text{ref},h}$, find \mathbf{u}^h , ϕ^h and λ^h such that

$$\sum_{j \in \text{U}_{\text{dof}}} K_{ij}(\phi^h) u_j = \lambda^h F_i^{\text{ref}}(\phi^h), \quad \forall i \in \text{U}_{\text{dof}}, \quad (55)$$

$$f_i(\mathbf{u}^h, \phi^h) \leq 0, \quad \dot{\phi}_i \geq 0, \quad f_i(\mathbf{u}^h, \phi^h) \dot{\phi}_i = 0, \quad \forall i \in \text{S}_M, \quad (56)$$

$$\max_{i \in \text{S}_M} \dot{\phi}_i = \dot{\phi}_{\text{max}}, \quad (57)$$

with initial condition at $t = 0$

$$\phi^h = 0 \text{ in } \Omega^h, \quad (58)$$

where

$$K_{ij}(\phi^h) = \int_{\Omega^h \setminus \Omega_c^h} \nabla \mathbf{L}_i : [(1 - d(\phi^h))\mathbf{C}] : \nabla \mathbf{L}_j \, d\omega, \quad \forall (i, j) \in \text{U}_{\text{dof}} \times \text{U}, \quad (59)$$

$$F_i^{\text{ref}}(\phi^h) = \int_{\Gamma_T^h} \mathbf{T}_d^{\text{ref},h} \cdot \mathbf{L}_i \, da - \sum_{j \in \text{U}_{\text{fixed}}} K_{ij}(\phi^h) \times (u_d^{\text{ref}})_j, \quad \forall i \in \text{U}_{\text{dof}}, \quad (60)$$

$$f_i(\mathbf{u}^h, \phi^h) = \langle \mathbf{Y}(\mathbf{u}^h), d'(\phi^h) \mathbf{M}_i \rangle - \langle \mathbf{Y}_c h(d(\phi^h)), d'(\phi^h) \mathbf{M}_i \rangle, \quad \forall i \in \text{S}_M. \quad (61)$$

4. Time discretization

In this section we give the prediction–correction scheme used to update both the damage field (through ϕ) and the displacement field. To simplify notations, we remove the h superscript from discretized fields. We assume a strictly increasing instant sequence $(t^n)_{n \in \mathbb{N}}$ with $t^0 = 0$ and $t^{n+1} = t^n + \Delta t$. We denote $f^n = f(t^n)$ and $\Delta f = \Delta t \dot{f}^n$. Furthermore, to simplify notation, we denote $f^0 = f^n$ and $f^1 = f^{n+1}$. Problem 2 is discretized in time to

Problem 3 (Space and Time Discretized Mechanical Problem). Given $\mathbf{u}_d^{\text{ref}}$ and $\mathbf{T}_d^{\text{ref}}$, knowing ϕ^0 , find \mathbf{u}^1 , ϕ^1 and λ^1 such that

$$\sum_{j \in \text{U}_{\text{dof}}^1} K_{ij}(\phi^1) u_j^1 = \lambda^1 F_i^{\text{ref}}(\phi^1), \quad \forall i \in \text{U}_{\text{dof}}^1, \quad (62)$$

$$f_i(\mathbf{u}^1, \phi^1) \leq 0, \quad \Delta \phi_i \geq 0, \quad f_i(\mathbf{u}^1, \phi^1) \Delta \phi_i = 0, \quad \forall i \in \text{S}_M^1, \quad (63)$$

$$\max_{i \in \text{S}_M^1} \Delta \phi_i = \Delta \phi_{\text{max}}. \quad (64)$$

This problem is split into prediction and correction steps. The predictor solution is denoted with a 1/2 upper index.

Problem 4 (Prediction). Given $\mathbf{u}_d^{\text{ref}}$ and $\mathbf{T}_d^{\text{ref}}$, knowing \mathbf{u}^0 and ϕ^0 , find $\delta \mathbf{u} = \mathbf{u}^{1/2} - \mathbf{u}^0$, $\delta \phi = \phi^{1/2} - \phi^0$ and $\delta \lambda = \lambda^{1/2} - \lambda^0$ such that

$$\sum_{j \in \text{U}_{\text{dof}}^0} K_{ij}(\phi^0) \delta u_j = \delta \lambda F_i^{\text{ref}}(\phi^0), \quad \forall i \in \text{U}_{\text{dof}}^0, \quad (65)$$

$$f_i(\mathbf{u}^0, \phi^0) + \bar{\alpha}_i^0 \delta\phi_i + \bar{\beta}_i^0(\delta\mathbf{u}) \leq 0, \quad \forall i \in S_M^0, \quad (66)$$

$$\delta\phi_i \geq 0, \quad \forall i \in S_M^0, \quad (67)$$

$$(f_i(\mathbf{u}^0, \phi^0) + \bar{\alpha}_i^0 \delta\phi_i + \bar{\beta}_i^0(\delta\mathbf{u}))\delta\phi_i = 0, \quad \forall i \in S_M^0, \quad (68)$$

$$\max_{i \in S_M^0} \delta\phi_i = \Delta\phi_{\max}, \quad (69)$$

where

$$\bar{\alpha}_i^0 = \langle -Y_c h'(d(\phi^0)) d'(\phi^0), d'(\phi^0) M_i^0 \rangle, \quad (70)$$

$$\bar{\beta}_i^0 : \delta\mathbf{u} \mapsto \langle \boldsymbol{\varepsilon}(\delta\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}^0), d'(\phi^0) M_i^0 \rangle \quad (71)$$

and h' is the first derivative of the hardening function h .

Problem 5 (Correction). Given $\mathbf{u}_d^{\text{ref}}$ and $\mathbf{T}_d^{\text{ref}}$, knowing $\phi^{1/2}$, find \mathbf{u}^1 , ϕ^1 and λ^1 such that

$$\phi^1 = \phi^{1/2}, \quad (72)$$

$$\sum_{j \in U_{\text{dof}}^1} K_{ij}(\phi^1) u_j^1 = \lambda^1 F_i^{\text{ref}}(\phi^1), \quad \forall i \in U_{\text{dof}}^1, \quad (73)$$

$$\max_{i \in S_M^1} f_i(\mathbf{u}^1, \phi^1) = 0. \quad (74)$$

Note that we can rewrite Eq. (65) as

$$\sum_{j \in U_{\text{dof}}^0} K_{ij}(\phi^0) u_j^{\text{ref},0} = F_i^{\text{ref}}(\phi^0), \quad \forall i \in U_{\text{dof}}^0, \quad (75)$$

$$\delta u_i = \delta \lambda u_i^{\text{ref},0}, \quad \forall i \in U_{\text{dof}}^0, \quad (76)$$

and Eq. (73) as

$$\sum_{j \in U_{\text{dof}}^1} K_{ij}(\phi^1) u_j^{\text{ref},1} = F_i^{\text{ref}}(\phi^1), \quad \forall i \in U_{\text{dof}}^1, \quad (77)$$

$$u_i^1 = \lambda^1 u_i^{\text{ref},1}, \quad \forall i \in U_{\text{dof}}^1, \quad (78)$$

from which we see that the linear system involved in the correction step at instant t^n is the same as the linear system involved in the prediction step at instant t^{n+1} . Therefore, this linear system is solved once and we obtain the \mathbf{u}^{ref} field. The solution to the prediction problem is then given by

$$\delta\lambda = \min_{i \in S_M^0} \left(\frac{f_i(\mathbf{u}^0, \phi^0) + \bar{\alpha}_i^0 \Delta\phi_{\max}}{-\bar{\beta}_i^0(\mathbf{u}^{\text{ref}})} \right), \quad (79)$$

$$\delta\mathbf{u} = \delta\lambda \mathbf{u}^{\text{ref}}, \quad (80)$$

$$\Delta\phi_i = \frac{\langle f_i(\mathbf{u}^0, \phi^0) + \bar{\beta}_i^0(\delta\mathbf{u}) \rangle_+}{-\bar{\alpha}_i^0}, \quad \forall i \in S_M^0. \quad (81)$$

The solution to the correction problem is given by

$$\lambda^1 = \sqrt{\min_{i \in S_M^1} \left(\frac{\bar{b}_i^1}{\bar{a}_i^1} \right)}, \quad (82)$$

$$\mathbf{u}^1 = \lambda^1 \mathbf{u}^{\text{ref}}, \quad (83)$$

where

$$\bar{a}_i^1 = \langle Y(\mathbf{u}^{\text{ref}}), d'(\phi^1) M_i^1 \rangle, \quad (84)$$

$$\bar{b}_i^1 = \langle Y_c h(d(\phi^1)), d'(\phi^1) M_i^1 \rangle. \quad (85)$$

5. Implementation

This section provides a description of the implementation in a finite element code. In the first part, we focus on modes computation and the ϕ field reinitialization and in the second part we describe the X-FEM enrichment using virtual nodes. All the algorithms are also detailed as flowcharts in the Appendix.

5.1. Modes computation and ϕ reinitialization

Modes are known as soon as the coefficients present in Eq. (48) are known. We place such coefficients in a matrix that we call the delocalization matrix

$$[M] = (M_{ij})_{\substack{i \in S_M \\ j \in S}}. \quad (86)$$

This matrix is initialized as the identity matrix with a number of lines corresponding to the number of nodes in the mesh ($\text{card}(S)$). At this stage, damage evolution is purely local. As non-locality steps in, the number of lines in the matrix reduces (since local modes are being aggregated) while the number of columns does not change. At any given step, matrix M can be expressed as

$$[M] = \begin{pmatrix} I_L & 0 & 0 \\ 0 & I_{\text{NLB}} & C \end{pmatrix}. \quad (87)$$

Local modes form the first lines of the matrix followed by the non-local modes. Matrix $[I_L]$ is the identity matrix of size $\text{card}(S_L)$, $[I_{\text{NLB}}]$ is the identity matrix of size $\text{card}(S_{\text{NLB}})$ and

$$[C] = (C_{ij})_{\substack{i \in S_{\text{NLB}} \\ j \in S_{\text{NLI}}}}, \quad (88)$$

is a rectangular matrix of size $\text{card}(S_{\text{NLB}}) \times \text{card}(S_{\text{NLI}})$.

Eq. (50) is computed using the Fast Marching Method (FMM). The FMM allows to build solutions of the two following boundary value problems, respectively called reinitialization problem and velocity extension problem. Knowing ϕ on Γ , find $\tilde{\phi}$ in Ω^+ such that

$$\|\nabla \tilde{\phi}\| = 1 \text{ in } \Omega^+, \quad (89)$$

$$\tilde{\phi} = \phi \text{ on } \Gamma, \quad (90)$$

and knowing $\tilde{\phi}$ in Ω^+ and v on Γ , find \tilde{v} in Ω^+ such that

$$\nabla \tilde{\phi} \cdot \nabla \tilde{v} = 0 \text{ in } \Omega^+, \quad (91)$$

$$\tilde{v} = v \text{ on } \Gamma. \quad (92)$$

The FMM algorithms are not described in this paper and can be found in detail in [28] and [29]. Only the input data and the exploitation of the output data are considered in the following. If we rewrite the velocity extension problem for modes, we have that, for all modes $i \in S_{\text{NLB}}$

$$\nabla \tilde{\phi}^h \cdot \nabla M_i = 0 \text{ in } \Omega^{+,h}, \quad (93)$$

$$M_i = N_i \text{ on } \Gamma^h. \quad (94)$$

Nodes belonging to S_{NLB} are located on the discrete interface Γ^h between the nonlocal zone and local zone which means that boundary condition can be written $\tilde{\phi}_j = \phi_j$ for the reinitialization problem and $M_{ij} = \delta_{ij}$ for the velocity extension problem, for all $j \in S_{\text{NLB}}$ and where δ_{ij} is the Kronecker symbol. For the reinitialization problem input and output data are

- input data: $\phi_i, i \in S_{\text{NLB}}$
- output data: $\phi_i, i \in S_{\text{NLI}}$.

For the velocity extension problem they are

- input data: $C_{ij} = \delta_{ji}, i \in S_{\text{NLB}}, j \in S_{\text{NLB}}$
- output data: $C_{ji}, i \in S_{\text{NLB}}, j \in S_{\text{NLI}}$.

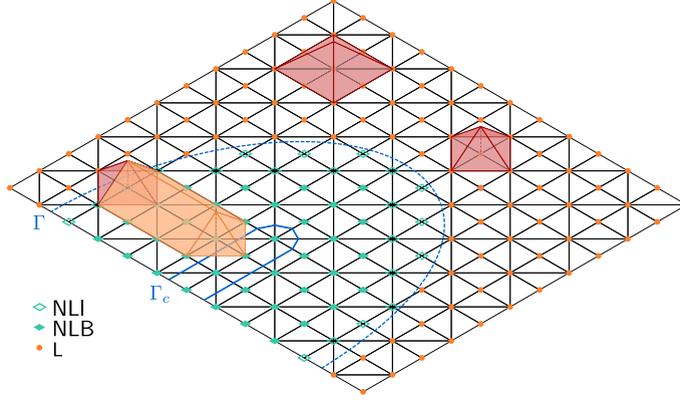


Fig. 4. Illustration of modes in local and nonlocal zones.

Fig. 4 illustrates modes. Modes in the local zone are simply nodal piecewise linear approximation functions whereas in the nonlocal zone, they are obtained by transports of nodal piecewise linear approximation functions taken at the boundary into the interior of the nonlocal zone. Once the delocalization matrix is built, computation of coefficients \bar{X}_i of any nonlocal field \bar{X}^h is done by two finite element assembling procedures. Given ϕ^h and X^h , for any test field $v^h = \sum_{i \in S} v_i N_i$ assemble the linear form

$$v^h \mapsto \int_{\Omega^h \setminus \Omega_c^h} X^h d'(\phi^h) v^h d\omega, \quad (95)$$

into vector $\{A\}$ and assemble the linear form

$$v^h \mapsto \int_{\Omega^h \setminus \Omega_c^h} d'(\phi^h) v^h d\omega, \quad (96)$$

into vector $\{B\}$. Then for $\{\bar{X}\} = (\bar{X}_i)_{i \in S_M}$

$$\{\bar{X}\} = ([M]\{A\}) \cdot / ([M]\{B\}), \quad (97)$$

where $\cdot /$ is the element-wise division.

5.2. Enrichment by virtual nodes

In this section, we present the enrichment considered in this work. This enrichment is interesting when used with a finite element code that does not provide an easy way to add additional degrees of freedom. The enrichment is added through virtual nodes as proposed in [25,26] and is equivalent to the Hansbo enrichment [24]. Note that the virtual node approach does not modify the obtained displacement field [30].

The approximation of the displacement field reads

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i \in U} u_i \mathbf{L}_i(\mathbf{x}), \quad (98)$$

where the set U will enlarge as damage grows (to account for displacement discontinuity). This approximation is a FEM approximation for which the underlying mesh evolves. Following the work of [26] on a mesh generation using virtual nodes, a hybrid mesh approach is considered. In such an approach, two meshes are required: the simulation mesh, that is the finite element mesh and the quadrature mesh. There are no degrees of freedom associated to the latter. In the context of this paper, the quadrature rule is a standard Gauss quadrature for uncut elements and a Gauss quadrature based on an element partition, as presented in [31], for cut elements. As a consequence, the quadrature mesh is built using a triangulation of cut elements only. Quadrature elements are classified either in fully damaged zone or nonlocal zone depending on values of ϕ compared to l_c . Fig. 5 illustrates such a quadrature mesh and describes

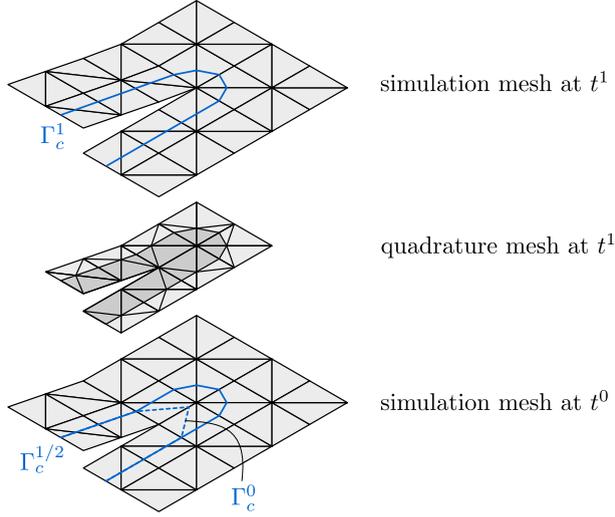


Fig. 5. Enrichment with virtual nodes as Ω_c evolves, $\Gamma_c^n = \text{iso-}l_c(\phi^n)$ (simulation meshes are elevated to see the duplication).

the simulation mesh update process: from bottom to top, we see a simulation mesh at an instant t^n over which the ϕ^n field has been updated to $\phi^{n+1/2}$, it is the state of computation after the prediction step going from t^n to t^{n+1} . The quadrature mesh is obtained from a triangulation of elements cut by $\Gamma_c^{n+1/2}$ on the simulation mesh at instant t^n . The simulation mesh is then updated by studying quadrature mesh connectivities. It is based on the following steps (further detailed in the Appendix):

- model cleaning: fully damaged nodes (i.e. nodes for which $d = 1$) and cut elements are removed from the simulation mesh, connectivity table is updated, related approximation coefficients are removed from approximations (but everything is kept in memory),
- node creation: new fully damaged nodes are duplicated from removed ones and are added to the mesh (to each removed node, one or several nodes with same location are added, connectivities differ),
- element creation: new cut elements are duplicated from corresponding new nodes and are added to the mesh,
- data transfer: new approximation coefficients (if any) on added entities are initialized to values of corresponding removed ones (several new nodes take value from the same removed node),
- ϕ reinitialization: ϕ_i new values are computed with a fast marching call by taking the neighboring nodes layer as boundary,
- deletion: removed objects are deleted.

6. Numerical applications

Two numerical experiments are considered. The first one is a pull-out test and the second one is the failure of specimen with an initial notch. For both cases, the same damage profile is used

$$d(\phi) = \begin{cases} 2(\phi/l_c) - (\phi/l_c)^2, & 0 \leq \phi/l_c \leq 1 \\ 1, & \phi/l_c \geq 1. \end{cases} \quad (99)$$

It is depicted in Fig. 1.

6.1. Pull out test

This test is anti-plane. The deformation is orthogonal to the sheet. The outer circle, Fig. 6, is clamped whereas the central cylinder is pulled up with a rigid displacement. The cross-section of the cylinder is either a disk, an ellipsis or a cross. All three cross-sections have the same perimeter. Lengths are made non-dimensional using the outer radius:

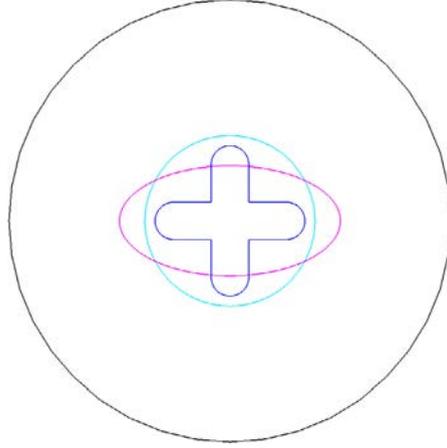


Fig. 6. Three geometries for the pull out test.

the outer radius is thus 1. The inner disk has a radius of 0.385, the ellipse radii are 0.5 and 0.25. Finally, the cross is formed of 8 straight segments with length 0.170 and four semi-circle of radius 0.0848. The TLS length l_c is set to 0.1. Stress type quantities are made non-dimensional using the Young modulus. The shear modulus (half Young's modulus assuming a Poisson ratio of 0) is thus 0.5. The critical energy release rate and softening function are given by

$$Y_c = 5.10^{-9}, \quad h(d) = \exp(4d). \quad (100)$$

With the above choice, the local damage models become unstable for the critical damage value of 0.75. Before this value, the model exhibits a hardening effect. Beyond this value, it softens.

The mesh size chosen is uniform and satisfies $h/l_c \sim 0.15$, whereas $\Delta\phi_{\max} = 0.99h$. Fig. 7 shows the total force needed with respect to the imposed displacement. The curve is obtained directly from the algorithm and no smoothing is applied. The observed slight oscillation may come from the explicit character of the algorithm as well as the lack of dissipation parameter in the algorithm.

It is observed that the cross gives the lowest failure force followed by the ellipse. The cross failure is less “brittle” than for the circle and does not show a snap-back phenomenon.

The development of damage may be studied in Fig. 8. For the circle, the damage develops in a uniform manner around its perimeter. The peak load is obtained for a value of damage close to the critical damage value. At this stage, the non-local damage zone is not yet formed. The damage then develops in a rather symmetrical manner. For the ellipse and the cross, damage develops first on the closest point to the outer circle (where the shear is maximum). At the peak load, some points are already fully damaged. After the peak load, damage progresses as well as the extent of the non-local zone. In the cross case, the crack goes does not follow the perimeter of the cross.

6.2. Notched specimen

The second test case is depicted in Fig. 9. The domain is loaded with imposed upward displacement under plane strain. The length is $H = 1m$ and the angle is $\theta = 45^\circ$. The Young modulus is set to 210 GPa and Poisson ratio to 0.3. The critical energy release rate and softening function are given by

$$Y_c = 1.3210^5 \text{ Pa}, \quad h(d) = \exp(4d). \quad (101)$$

The TLS length l_c is set to 0.1 m. The mesh size, h , is finer along the expected crack path in order to satisfy $h/l_c = 0.1$.

The load displacement curve is given in Figs. 10 and 11 depicts the evolution of damage. It may be seen that damage is already developed at the peak load (this is not surprising since the damage evolution model has first a hardening effect). Snap-back only occurs at half the peak load.

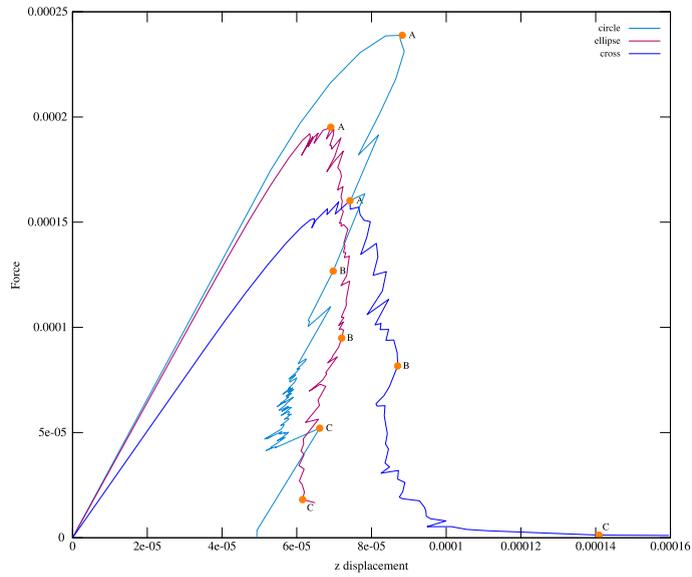


Fig. 7. Load–displacement curves for the three geometries. The load unit is Newton per meter (thickness) and the displacement is in meters.

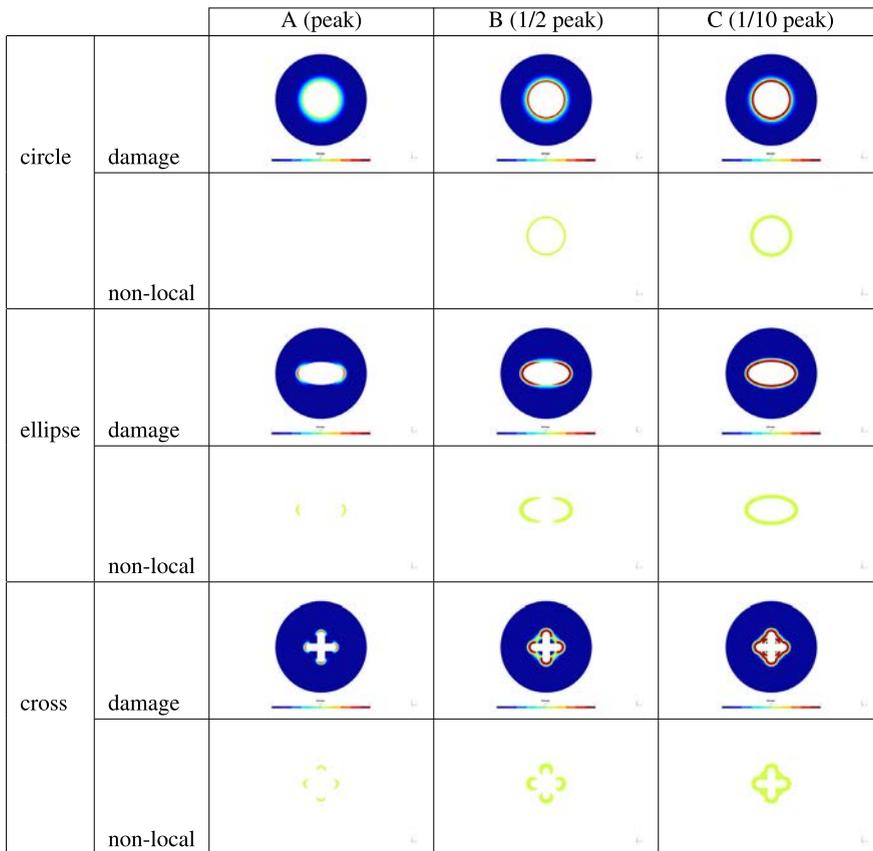


Fig. 8. Damage field and extent of the non-local zone for the three geometries at different loadings.

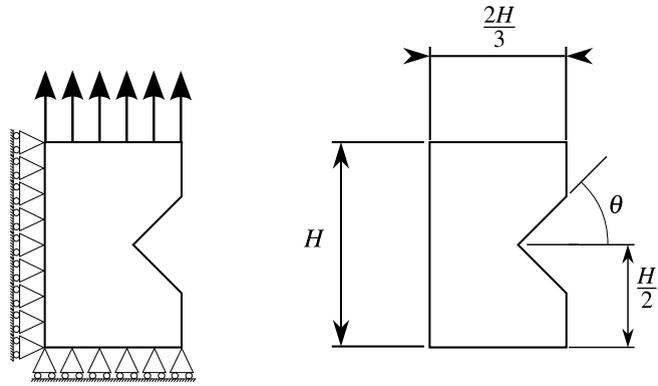


Fig. 9. Notch geometry.

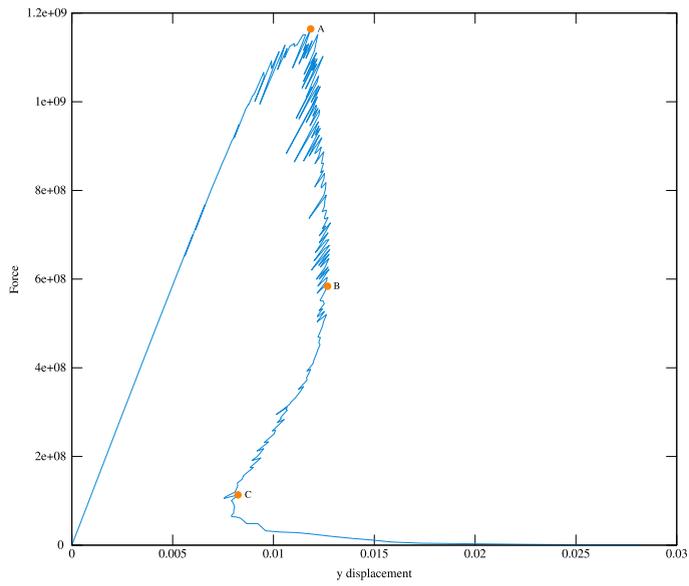


Fig. 10. Load–displacement curve for the notch geometry.

	A (peak)	B (1/2 peak)	C (1/10 peak)
damage			
non-local			

Fig. 11. Notch problem: damage and extent of the non-local zone at three different loadings (shown in the previous figure).

7. Conclusions

We have presented in this paper a new implementation of the Thick Level Set approach. Both local and non-local damage may evolve in a concurrent manner. In the implementation chosen, an element is either in the local or non-local zone. This simplifies the set up on a computer. Displacement discontinuity is inserted by element duplication. This allows to use legacy code not able to handle easily a varying number of degrees of freedom per element. From the computational point of view, we also introduced a way to update damage without any matrix solve.

Regarding future work, extension to 3D could obviously be investigated. It does not raise specific difficulties. We are also aware of the fact that a bifurcation analysis would be useful. The definition of the local and non-local zones is only based on the evolution of the damage gradient. It does not guarantee that every point in the local zone is stable (positive definite stiffness tangent matrix).

This analysis could force meta-stable locus in the local zone to be placed in the non-local zone.

Acknowledgments

Olivier Pierard is acknowledged for proofreading the paper. The authors gratefully acknowledge the support of the ERC advanced grant XLS No. 291102.

Appendix. Algorithms

In this appendix, we describe algorithms in a compact manner. All of the material concerning enrichment with virtual nodes is given in a separate box to keep a maximum orthogonality between parts of the algorithm. (See Figs. A.12–A.14.)

Given: $\mathbf{u}_d^{\text{ref}}$ and $\mathbf{T}_d^{\text{ref}}$
 Initial state: $\{\phi\} = \{0\}$ (i.e. $\phi = 0$), $[M] = (\delta_{ij})_{\substack{i \in S \\ j \in S}}$,
 $S_M = S_L = S$ and $S_{NLB} = S_{NLI} = \emptyset$.

1. assemble $[K(\phi)]$ and $\{F^{\text{ref}}(\phi)\}$,
2. solve $[K(\phi)]\{u^{\text{ref}}\} = \{F^{\text{ref}}(\phi)\}$,
3. compute $\{\bar{a}\}$ and $\{\bar{b}\}$ (alg. 2),
4. compute load factor λ (alg. 4),
5. post process, exit if $\lambda \max(\{u^{\text{ref}}\})$ (maximum displacement) is too big or increase step number,
6. compute $\{\bar{\alpha}\}$ and $\{\bar{\beta}\}$ (alg. 3),
7. compute load factor variation $\delta\lambda$ (alg. 5),
8. update ϕ field (alg. 6),
9. classify nodes (alg. 7),
10. reinitialize ϕ field (alg. 8),
11. update X-FEM enrichment (alg. 10),
12. compute $[M]$ (alg. 9),
13. goto 1.

Algorithm 1: general algorithm.

Knowing ϕ , \mathbf{u}^{ref} and $[M]$:

1. assemble linear form $v \mapsto \int_{\Omega \setminus \Omega_c} Y(\mathbf{u}^{\text{ref}}) d'(\phi) v \, d\omega$ into vector $\{a\}$,
2. assemble linear form $v \mapsto \int_{\Omega \setminus \Omega_c} Y_c h(d(\phi)) d'(\phi) v \, d\omega$ into vector $\{b\}$,
3. compute $\{\bar{a}\} = [M]\{a\}$,
4. compute $\{\bar{b}\} = [M]\{b\}$.

Return $\{\bar{a}\}$ and $\{\bar{b}\}$.

Algorithm 2: nonlocal computation of constitutive model (for load factor computation).

Knowing λ , ϕ , \mathbf{u}^{ref} (at integration points) and $[M]$:

1. assemble linear form $v \mapsto \int_{\Omega \setminus \Omega_c} Y_c h'(d(\phi)) (d'(\phi))^2 v \, d\omega$ into vector $\{\alpha\}$,
2. assemble linear form $v \mapsto \int_{\Omega \setminus \Omega_c} 2\lambda Y(\mathbf{u}^{\text{ref}}) v d'(\phi) \, d\omega$ into vector $\{\beta\}$,
3. compute $\{\bar{\alpha}\} = [M]\{\alpha\}$,
4. compute $\{\bar{\beta}\} = [M]\{\beta\}$.

Return $\{\bar{\alpha}\}$ and $\{\bar{\beta}\}$.

Algorithm 3: nonlocal computation of constitutive model (for load factor variation computation).

Knowing $\{\bar{a}\}$, $\{\bar{b}\}$:

1. compute $\lambda^2 = \min(\{\bar{b}\} \cdot / \{\bar{a}\})$,
2. compute $\{\bar{f}\} = \lambda^2 \{\bar{a}\} - \{\bar{b}\}$,
3. compute $\lambda = \sqrt{\lambda^2}$.

Return λ and $\{\bar{f}\}$.

Algorithm 4: load factor computation.

Knowing $\{\bar{f}\}$, $\{\bar{\alpha}\}$, $\{\bar{\beta}\}$:

1. compute $\delta\lambda = \min[(\Delta\phi_{\max} \{\bar{\alpha}\} - \{\bar{f}\}) \cdot / \{\bar{\beta}\}]$,
2. compute $\{\Delta\phi\} = \max(\{0\}, \{\bar{f}\} + \delta\lambda \{\bar{\beta}\}) \cdot / \{\bar{\alpha}\}$,

Return $\delta\lambda$ and $\{\Delta\phi\}$.

Algorithm 5: load factor variation computation.

Knowing $\{\phi\}$, $\{\Delta\phi\}$ and $[M]$:

1. compute $\{\phi\} += [M]^T \{\Delta\phi\}$.

Algorithm 6: ϕ update.

Knowing ϕ :

1. search for finite elements for which $\|\nabla\phi\| \geq 1$,
2. collect node numbers in S_{NL} ,
3. compute $S_L = S \setminus S_{NL}$,
4. compute S_{NLB} : loop on S_{NL} nodes and search for a S_L neighbor,
5. compute $S_{NLI} = S_{NL} \setminus S_{NLB}$,
6. compute $S_M = S_L \cup S_{NLB}$.

Return all sets.

Algorithm 7: nodal classification.

Knowing ϕ , S_{NLB} and S_{NLI} :

1. call Fast Marching with input data: for all $i \in S_{NLB}$, give ϕ_i ,
2. update ϕ with output data: for all $i \in S_{NLI}$, obtain ϕ_i ,

Return ϕ .

Algorithm 8: ϕ reinitialization.

Knowing ϕ , S_{NLB} and S_{NLI} :

1. call Fast Marching with input data: for all $i \in S_{NLB}$, give $\{V\}_i = (\delta_{ji})_{j \in S_{NLB}}$,
2. build $[M]$ with output data: for all $i \in S_{NLI}$, obtain $\{V\}_i = (M_{ji})_{j \in S_{NLB}}$,

Return ϕ .

Algorithm 9: $[M]$ computation.

Knowing ϕ and the simulation mesh:

1. generate the quadrature mesh (triangulation of cut elements only),
2. remove nodes for which $\phi > l_c$ and remove adjacent elements,
3. dispatch neighbor nodes (alg. 11),
4. duplicate removed nodes (alg. 12),
5. duplicate removed elements (alg. 13),
6. transfer data (if any) from removed entities to added entities,
7. call Fast Marching with input data: for all neighbor nodes, give ϕ value,
8. update ϕ with output data: for all duplicated nodes, obtain ϕ value,

Return the simulation mesh and the quadrature mesh.

Algorithm 10: X-FEM enrichment using virtual nodes.

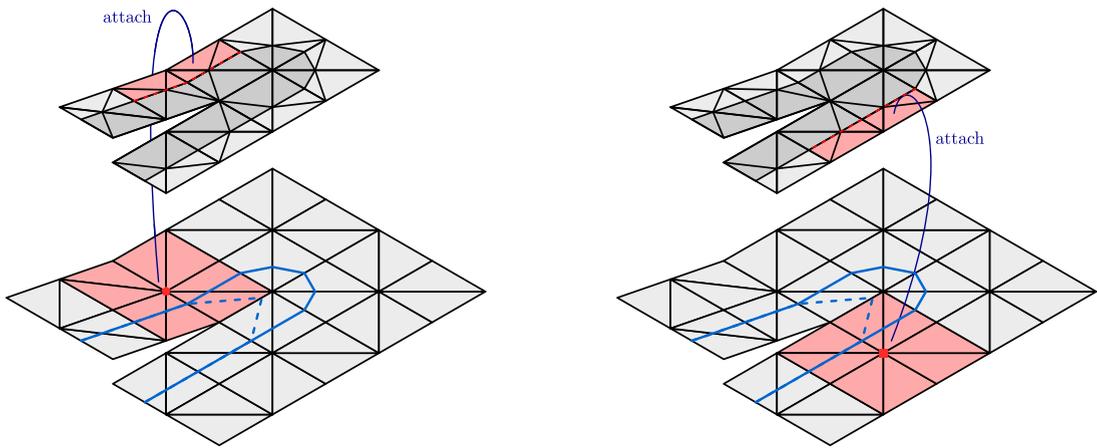


Fig. A.12. Neighbor nodes dispatch.

For all neighbor nodes,

1. collect adjacent elements,
2. collect quadrature elements from adjacent elements,
3. find connected components,
4. keep only the connected component in nonlocal zone,
5. attach node to the connected component.

Algorithm 11: neighbor nodes dispatch.

For all removed nodes,

1. collect adjacent elements,
2. collect quadrature elements from adjacent elements,
3. find connected components,
4. keep only connected components in nonlocal zone,
5. duplicate node: a node per connected component,
6. attach duplicated node to the connected component.

Algorithm 12: Removed nodes duplication.

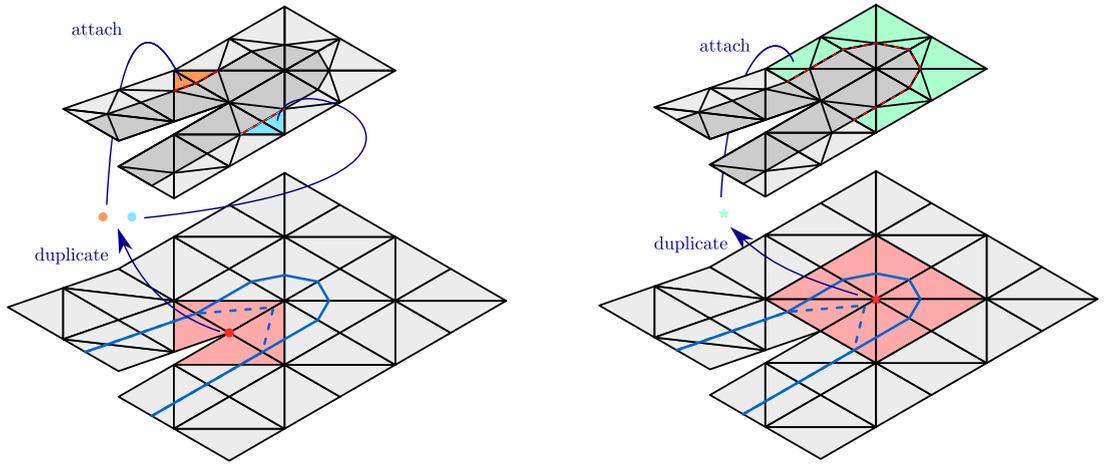


Fig. A.13. Removed nodes duplication.

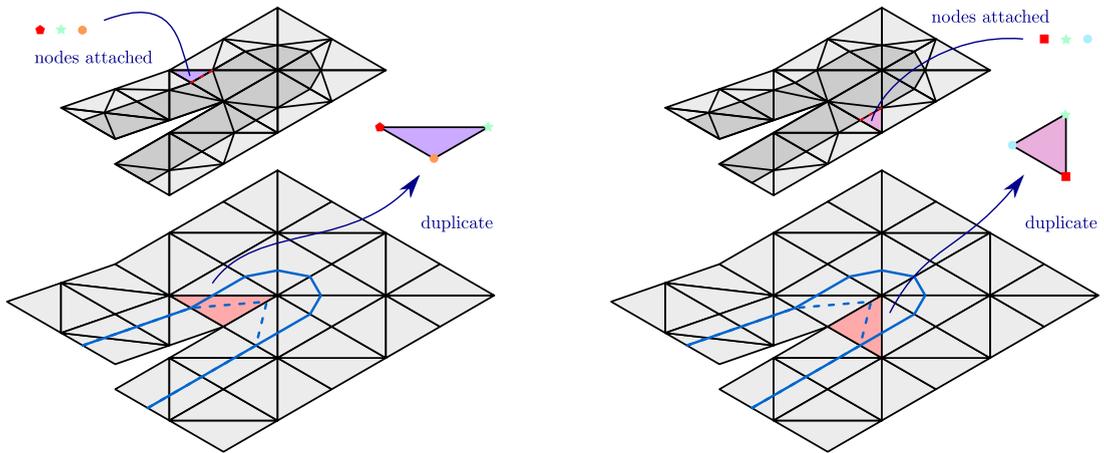


Fig. A.14. Removed elements duplication.

For all removed elements,

1. collect quadrature elements,
2. find connected components,
3. keep only connected components in nonlocal zone,
4. collect attached nodes,
5. duplicate element using collected nodes.

Algorithm 13: removed elements duplication.

References

- [1] D.S. Dugdale, Yielding of steel sheets containing slits, *J. Mech. Phys. Solids* 8 (2) (1960) 100–104.
- [2] G.I. Barenblatt, The mathematical theory of equilibrium cracks in brittle fracture, *Adv. Appl. Mech.* 7 (1) (1962) 55–129.
- [3] A. Hillerborg, M. Mod er, P.-E. Petersson, Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements, *Cement and Concrete Res.* 6 (6) (1976) 773–781.
- [4] J. Lemaitre, J.-L. Chaboche, A. Benallal, R. Desmorat, *M canique Des Mat riaux Solides-3eme  dition*, Dunod, 2009.
- [5] J.-J. Marigo, Formulation d’une loi d’endommagement d’un mat riaux  lastique, *C. R. Acad. Sci., Paris II* 292 (1981) 1309–1312.
- [6] J.-C. Simo, J.-W. Ju, Strain-and stress-based continuum damage models—i. formulation, *Int. J. Solids Struct.* 23 (7) (1987) 821–840.
- [7] T. Belytschko, D. Lasry, A study of localization limiters for strain-softening in statics and dynamics, *Comput. Struct.* 33 (3) (1989) 707–715.
- [8] Z. Bazant, T. Belytschko, T.-P. Chang, Continuum theory for strain-softening, *J. Eng. Mech.* 110 (12) (1984) 1666–1692.
- [9] G. Pijaudier-Cabot, Z.P. Bazant, Nonlocal damage theory, *J. Eng. Mech.* 113 (10) (1987) 1512–1533.
- [10] A. Needleman, Material rate dependence and mesh sensitivity in localization problems, *Comput. Methods Appl. Mech. Engrg.* 67 (1) (1988) 69–85.
- [11] O. Allix, J.-F. De , Delayed-damage modelling for fracture prediction of laminated composites under dynamic loading, *Eng. Trans.* (1997) 29–46.
- [12] N. Triantafyllidis, E. Aifantis, A gradient approach to localization of deformation. I. Hyperelastic materials, *J. Elasticity* 16 (1986) 225–237. <http://dx.doi.org/10.1007/BF00040814>. <http://www.springerlink.com/index/N66Q75208686G464pdf>.
- [13] R. Peerlings, R. De Borst, W. Brekelmans, J. Vree, Gradient-enhanced damage for quasi-brittle materials, *Int. J. Numer. Methods Engrg.* 39 (1996) 3391–3403.
- [14] B. Bourdin, G.a. Francfort, J.-J. Marigo, The variational approach to fracture, *J. Elasticity* 91 (1–3) (2008) 5–148. <http://dx.doi.org/10.1007/s10659-007-9107-3>.
- [15] V. Hakim, A. Karma, Laws of crack motion and phase-field models of fracture, *J. Mech. Phys. Solids* 57 (2) (2009) 342–368. <http://dx.doi.org/10.1016/j.jmps.200810012>. <http://linkinghub.elsevier.com/retrieve/pii/S0022509608001774>.
- [16] R. de Borst, C.V. Verhoosel, Gradient damage vs phase-field approaches for fracture: Similarities and differences, *Comput. Methods Appl. Mech. Engrg.* <http://dx.doi.org/10.1016/j.cma.2016.05.015>. <http://linkinghub.elsevier.com/retrieve/pii/S0045782516303796>.
- [17] A. Simone, G.N. Wells, L.J. Sluys, From continuous to discontinuous failure in a gradient-enhanced continuum damage model, *Comput. Methods Appl. Mech. Engrg.* 192 (41) (2003) 4581–4607.
- [18] C. Comi, S. Mariani, U. Perego, An extended fe strategy for transition from continuum damage to mode i cohesive crack propagation, *Int. J. Numer. Anal. Methods Geomech.* 31 (2) (2007) 213–238.
- [19] E. Tamayo-Mas, A. Rodr guez-Ferran, A medial-axis-based model for propagating cracks in a regularised bulk, *Internat. J. Numer. Methods Engrg.* 101 (7) (2014) 489–520. <http://dx.doi.org/10.1002/nme.4757>. <http://arxiv.org/abs/1010.1724>.
- [20] N. Mo s, C. Stolz, P.-E. Bernard, N. Chevaugeon, A level set based model for damage growth: the thick level set approach, *Internat. J. Numer. Methods Engrg.* 86 (3) (2011) 358–380.
- [21] N. Mo s, C. Stolz, N. Chevaugeon, Coupling local and non-local damage evolutions with the thick level set model, *Adv. Modeling Simul. Eng. Sci.* 1 (1) (2014) 1–21.
- [22] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Science*, Cambridge University Press, 1999.
- [23] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, 2003.
- [24] A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 193 (33) (2004) 3523–3540.
- [25] J.-H. Song, P.M.A. Areias, T. Belytschko, A method for dynamic crack and shear band propagation with phantom nodes, *Internat. J. Numer. Methods Engrg.* 67 (6) (2006) 868–893.
- [26] C.L. Richardson, J. Hegemann, E. Sifakis, J. Hellrung, J.M. Teran, An xfem method for modeling geometrically elaborate crack propagation in brittle materials, *Internat. J. Numer. Methods Engrg.* 88 (10) (2011) 1042–1065.
- [27] K. Moreau, N. Mo s, D. Picart, L. Stainier, Explicit dynamics with a non-local damage model using the thick level set approach, *Internat. J. Numer. Methods Engrg.* 102 (3–4) (2015) 808–838.
- [28] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Natl. Acad. Sci.* 93 (4) (1996) 1591–1595.
- [29] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, *J. Comput. Phys.* 148 (1) (1999) 2–22.
- [30] P.M.A. Areias, T. Belytschko, A comment on the article“a finite element method for simulation of strong and weak discontinuities in solid mechanic” by a. hansbo and p. hansbo [*comput. methods appl. mech. engrg.* 193 (2004) 3523–3540], *Comput. Methods Appl. Mech. Engrg.* 195 (9) (2006) 1275–1276.
- [31] N. Mo s, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *Internat. J. Numer. Methods Engrg.* 46 (1) (1999) 131–150.