

# Python Scripting based Implementation of Phase Field Method in Abaqus

<sup>a</sup>Anshul Pandey, <sup>a</sup>Sachin Kumar\*

<sup>a</sup>*Department of Mechanical Engineering, Indian Institute of Technology Ropar,  
Rupnagar, Punjab, India – 140001*

*\*Email: [sachin@iitrpr.ac.in](mailto:sachin@iitrpr.ac.in)*

## ABSTRACT

The phase field models can coherently address the relative complex fracture phenomenon such as crack nucleation, branching, deflection, etc. The phase field method has been implemented in finite element package Abaqus for solving fracture problems in recent studies. The implementation of it is quite computationally expensive due to the requirement of high density mesh region in the anticipated fracture zone. Therefore, in the current work, a python scripting based mesh refinement strategy for phase field method (PFM) implementation in Abaqus, is presented. The proposed mesh refinement algorithm utilizes the Abaqus built-in posteriori error indicator MISESERRI to track the crack path and demarcate the regions require mesh refinement. The python script enables the simultaneous execution of pre-assessment of crack path, local mesh refinement and phase field method implementation, altogether in a single pass. The accuracy and effectiveness of the proposed algorithm is validated through several standard fracture problems. The proposed mesh refinement algorithm based PFM results demonstrate significant reduction in computational time and savings in memory resource requirement as compared to the standard phase field method, which are quite encouraging.

*Keywords - Phase field fracture, Mesh refinement, Finite element method, Abaqus, UEL subroutine, Python scripting*

## I. INTRODUCTION

Fracture plays a crucial role in many engineering domains when it comes to keeping structures functional under varied loading conditions. It is one of the most prevalent modes of failure of engineering materials and structures. It has a substantial economic effect as avoiding failure owing to induced crack is a fundamental limitation in the design procedure for any engineering structure. Generally, it is achieved by incorporating significant safety factors throughout the design phase. For research community, the ability to predict brittle failure while considering its catastrophic effects, onset, and mechanism of propagation is crucial. Griffith developed the theory to investigate the evolution of brittle cracks [1], which was followed by Irwin's technique to account for high stresses and microscopic plasticity close to the crack tip by introducing stress intensity factor (SIF) for different modes of fracture failure popularly known as the theory of linear elastic fracture mechanics (LEFM) [2].

It appears to be extremely difficult to anticipate the failure of engineering structures due to fracture using analytical and experimental approaches, as the latter is economically unsound while solutions for the former are rare. Computational fracture modeling is an effective method for understanding and estimating the failure of engineering structures. The fracture of solids can be mathematically modeled using either a discontinuous (discrete) approach or a continuous (smeared/diffuse) approach. The displacement field is considered to be discontinuous at the crack surface in the discrete methods, whereas the stress field is generally degraded to describe the fracture process in the continuous approaches.

Several theories, like LEFM and cohesive zone model (CZM) [3,4], which fall within the category of discrete approach, have been developed in the past to handle the modeling of fracture phenomena through numerical techniques, such as the finite element method (FEM) [5], meshfree methods [6,7], extended finite element method (XFEM) [8,9], floating node method [10,11], and isogeometric analysis (IGA) [12,13]. These discontinuous approaches pose a challenge to model fracture nucleation, propagation, and direction as they require additional criteria. In the case of a mesh-based methodology, such as the finite element method which implements a continuous displacement field, introducing displacement discontinuity where the domain of the solution changes with time is quite tedious and extortionate. Although, the XFEM with enrichment functions is used as an alternative method to model the crack domain evolving with time [14], the issues of crack branching, merging, and intersecting cracks remain a challenging task for researchers, especially in three dimensions. The

cracking particle method (CPM) is an interesting alternative methodology that has been presented in literature for addressing complex fracture problems [15,16], where the crack is treated as a coalescence of discrete cracks that are constrained on nodes known as cracked particles. The approach is quite intuitive and efficient, and it can effectively replicate crack branching patterns in complex fracture problems. Peridynamic theory by Silling et al [17–19] has also been developed to address the discontinuous problems, where the interaction forces within the material are defined on the basis of a parameter referred to as horizon. Although, the discrete fracture approaches such as CPM, peridynamics etc. are able to qualitatively deal with the complex fracture problems yet the challenges associated with the discontinuous approaches prompted the development of continuous/smearred crack approaches, in which the displacement discontinuity or jump is regularized over a small localization band of finite width, enabling crack paths to be naturally ascertained as part of the solution. The continuum damage mechanics (CDM) is the most widely used theory under the continuous approach which establishes the effect and evolution of micro-cracks and micro-defects using damage variables [20,21]. The phase field method (PFM) is another recently developed approach as an alternative to CDM.

The phase field method has been developed as an alternative in recent years to address the issues of discrete or discontinuous approaches for modeling fracture problems. In PFM, the discontinuity in the form of a crack is modeled as damage/field smeared over a regularized region. The physics-based models derived from the Ginsberg-Landau phase evolution [22,23] and mechanics-based models derived from the variational theory of fracture [24] have been developed for modeling the fracture, where the latter helped to overcome the shortcomings of Griffith's approach to model the crack nucleation, deflection, branching, etc. The variational approach incorporates the minimization of the total potential energy with respect to the displacement field and a scalar phase variable governing the evolution of crack topology [25–31]. In order to assure distinct behavior in tensile and compressive fields, an additive decomposition of elastic energy density based on volumetric and deviatoric contributions has been presented in the literature [32,33]. Miehe's thermodynamically compatible phase field model based on principles of continuum mechanics and thermodynamics with anisotropic strain energy density split is an essential addition to phase-field modeling [34,35]. It incorporates crack irreversibility by inducing a local history field variable. A similar anisotropic model with a positive-negative decomposition of effective stress tensor has been presented by Wu et al [36,37]. Readers interested in a more compendious review are advised for the literature [38,39].

To establish the length scale parameter, phase field implementation requires a quite fine mesh, which makes it computationally intensive [38,40,41]. The mesh refinement in the domain where the crack growth is expected seems to be a precondition for the finite element implementation of the phase field method. To overcome the prerequisite of refined mesh, several adaptive mesh refinement strategies for phase-field models have been documented [42,43], where the critical value of the phase field serves as an element diagnostic for mesh refinement criteria. Klinsmann et al [44] proposed an adaptive phase field model that utilizes the gradient of phase field alongwith tensile part of maximum strain energy density as element diagnostic for mesh refinement. Heister et al [45] presented a predictor-corrector mesh refinement strategy based on threshold value of phase field. The crack irreversibility is imposed based on primal dual active set algorithm. The coarse elements are refined for preset maximum mesh refinement level, in the current time step. For adaptive phase field models, the effective crack driving force is also utilized as the mesh refinement criterion [46]. A much more efficient anisotropic adaptive model, where the mesh refinement is administered normally to crack propagation has been proposed by Artina et al [47]. Patil et al [48] proposed a coupled multiscale FEM with the phase field method for adaptive refinement, that involves the localized region in the proximity of diffuse crack to be discretized employing multiscale basis functions and modeling discontinuities with XFEM. Adaptive phase field models are also reported along with other numerical approaches such as scaled boundary finite element method [49], polygonal finite element method [50,51], posteriori error estimation [52], etc. In the literature [53], an alternate strategy for improving computational efficiency is presented, namely the use of physics informed neural network algorithms for the phase field method, which utilizes the variational energy of the system as the loss function to train the network.

The implementation of phase field method to study fracture problems has been done on several open-source software platforms like 'FENICS' [54], 'JIVE' [55], etc. Bourdin has presented a Fortran-based code MEF90, for the implementation of the phase field method. Moreover, adaptive phase field implementations based on FENICS, MATLAB [48,50], and JIVE [44] have been reported in the literature, which requires additional codes to develop nonlinear solvers such as Newton-Raphson iterative solver, Jacobian free Newton-Krylow solver [56], etc. On the other hand, the use of commercial finite element software such as Abaqus' subroutine utilities to implement the phase field model has been reported in the literature. The Abaqus phase field implementation requires a Fortran code in the form of subroutines without requirement of separate code for the solvers. The nonlinear solver of Abaqus has built-in support for the monolithic as well as staggered Newton-Raphson iterative scheme, modified Newton or BFGS [57] scheme, the latter requiring fewer iterations than the staggered Newton-Raphson scheme for phase field implementation. The Fortran subroutines in Abaqus denoted as UEL, UMAT, and HETVAL, can be used independently or in tandem with each other for phase field implementation.

A combination of UEL and UMAT subroutines to simulate and visualize brittle fracture problems employing a phase field model was reported by Msekhe et al [58]. The element level calculations are prescribed in the UEL subroutine while the UMAT subroutine is used for the post-processing.

Further building up with this conjuncture, Molnar et al [59] presented a layered system of user elements to implement a phase field model. Utilizing the analogy between the heat transfer partial differential equation and phase field force balance equation, the implementation of UMAT-HETVAL subroutines is presented in the literature [40]. Recently Wu and Huang [60] presented three unique implementations of the phase field method in Abaqus using UMAT and UEL. Other procedures for implementation in commercial finite element software Abaqus have been discussed in the literature as well [61–64].

The implementation of phase field method in Abaqus requires defining a region of low mesh size where the crack growth is expected. This can be significantly achieved based on the prior knowledge of crack path either via experimental methods or previous studies in literature. Even if the expected path of crack growth is known from the experimental/literature studies for simplistic problems such as Mode I and Mode II loading scenarios, it is insufficient to precisely work out the region where the mesh size can be reduced enough to fix the length scale parameter for capturing the phase field. Thus, a broad region with high mesh density is normally assigned which leads to large number of elements in the discretized domain. This situation augments in the case of complex fracture problems where multiple voids and/or notches are present in the specimen. Furthermore, for practical real-world fracture problems where the location of crack nucleation and subsequent growth is unknown, the study of such problems would impose significantly high order of the tangent stiffness matrix (AMATRX) for the analysis. The implementation of the phase field technique in Abaqus thus demands sufficiently large memory resources and produces a lengthy computational time. The adaptive mesh refinement approach allows for a decrease in computing resources while requiring significantly less computational time. Although, the use of adaptivity for phase field models have been reported in the literature yet the implementation of such adaptive phase field models in Abaqus have not been cited in the literature. An alternative strategy for obtaining reasonable computational time is to precisely redefine the zone of expected crack growth, that is characterised by a robust combination of an element diagnostic, such as the built-in error indicator in Abaqus, with the phase field function.

Therefore, the current study presents a methodology for implementing mesh refinement strategy in Abaqus to model fracture problems with the phase field method. The goal is accomplished by writing a python script for the Abaqus Python Development Environment (PDE) interface that integrates Abaqus' adaptive remeshing tool with subroutine utility. The python script includes methods or functions for developing the problem model, user material data, and mesh in the initial '\*.inp' file. The python script makes it easy to initialize the remeshing rule on a set of elements using the posteriori-error indicator MISESERRI. The '\*\_UEL.inp' file for UEL implementation of the phase field method is incorporated by a separate function in the script. The whole process is automated with the help of the aforementioned python script itself. The results of the phase field method with proposed mesh refinement algorithm are verified thereafter.

The chapter is organized as follows; an overview of the phase field method is presented in **Section 2**. The details of the implementation of the proposed mesh refinement algorithm for the phase field method are presented in **Section 3**. **Section 4** addresses the robustness of the proposed approach through numerical simulations and their outcomes, which is followed by conclusive statements in **Section 5**.

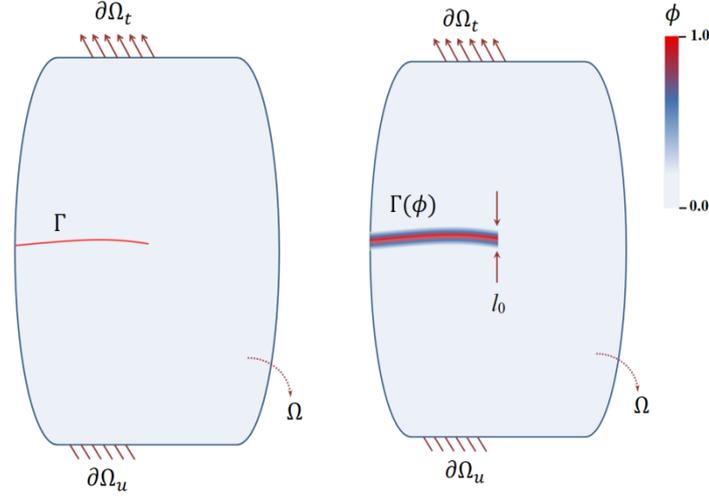
## II. PHASE-FIELD MODEL FOR FRACTURE

Consider a linear elastic solid continuum defined as the domain ( $\Omega$ ) of an  $N$ -dimensional space such that,  $\Omega \subset \mathbb{R}^{N_{dim}}$  and bounded by surface  $\partial\Omega^{N_{dim}-1}$ , ( $N_{dim} \in \{1,2,3\}$ ) as shown in **Fig. 1**. The displacement vector ( $\bar{\mathbf{u}}$ ) and traction vector ( $\bar{\mathbf{t}}$ ) are specified on  $\partial\Omega_u$  and  $\partial\Omega_t$  respectively, such that  $\partial\Omega_u \cap \partial\Omega_t = \emptyset$  &  $\partial\Omega_u \cup \partial\Omega_t = \Omega$ . The displacement field ( $\mathbf{u}$ ) and variation in displacement field ( $\delta\mathbf{u}$ ) are defined as,

$$\mathbf{u}(\mathbf{x}) \in \mathcal{S}_{\mathbf{u}}\{\mathbf{u} \mid \mathbf{u}(\mathbf{x}) = \bar{\mathbf{u}} \quad \forall \quad \mathbf{x} \in \partial\Omega_u\} \quad (1)$$

$$\delta\mathbf{u}(\mathbf{x}) \in \mathcal{S}_{\delta\mathbf{u}}\{\delta\mathbf{u} \mid \delta\mathbf{u}(\mathbf{x}) = 0 \quad \forall \quad \mathbf{x} \in \partial\Omega_u\} \quad (2)$$

The crack  $\Gamma(\phi) \subset \mathbb{R}^{N_{dim}-1}$ , with unit normal vector  $\hat{\mathbf{n}}_b$ , emerges once the failure criterion's limit is exceeded. The crack phase field ( $\phi$ ) topology can be characterized in localization band  $\mathcal{X} \subseteq \Omega$  in the vicinity of crack surface  $\Gamma(\phi)$  as,



**Figure 1: Schematic of a solid continuum representing an internal crack as (a) sharp and (b) diffuse**

$$\phi(\mathbf{x}) \in \mathcal{S}_\phi \{ \phi \mid \phi(\mathbf{x}) \in [0,1]; \dot{\phi}(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} \} \quad (3)$$

$$\delta\phi(\mathbf{x}) \in \mathcal{S}_{\delta\phi} \{ \delta\phi \mid \delta\phi(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \mathcal{X}; \delta\phi(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \Gamma \} \quad (4)$$

Griffith's thermodynamic framework serves as the foundation for the phase field fracture model. According to the first rule of thermodynamics, a crack can grow only if the total energy of the system decreases or remains constant. As a result of the assumption of equilibrium circumstances, i.e. no net change in total energy - a critical condition for a fracture may be formulated. An energy dissipation functional based on the principle of virtual work can be expressed as [65],

$$\mathcal{D}^c(\bar{\mathbf{u}}, \phi) = \mathcal{W}_{ext} - \mathcal{W}_{int} \quad (5)$$

where the internal work ( $\mathcal{W}_{int}$ ) consists of internal bulk energy and energy dissipation due to crack growth. The minimization of total potential energy functional is considered such that the variation in energy dissipation functional must follow  $\delta\mathcal{D}^c \geq 0$ . In consideration of minimization of the potential energy functional for any feasible deformation process, including crack evolution, the variational, ' $\delta\mathcal{D}^c$ ', may well be expressed as follows,

$$\delta\mathcal{D}^c = \delta\mathcal{W}_{ext} - \delta\mathcal{W}_{int} \geq 0 \quad (6)$$

The external work ( $\mathcal{W}_{ext}$ ) for prescribed body force vector ( $\bar{\mathbf{b}}$ ) and boundary traction vector ( $\bar{\mathbf{t}}$ ) on the reference configuration can be formulated as,

$$\delta\mathcal{W}_{ext} = \int_{\Omega} \bar{\mathbf{b}} \cdot \delta\mathbf{u} \, dV + \int_{\partial\Omega} \bar{\mathbf{t}} \cdot \delta\mathbf{u} \, dA \quad (7)$$

To couple the crack evolution with the deformation process, the associated internal work is expressed as a confluence of internal bulk energy ( $\delta\Psi^b$ ) and dissipated energy ( $\delta\Psi^\Gamma$ ) due to the evolution of the crack,

$$\delta\mathcal{W}_{int} = \int_{\Omega} \delta\Psi^b(\boldsymbol{\epsilon}, \phi) \, dV + \int_{\Omega} \delta\Psi^\Gamma(\phi) \, dV \quad (8)$$

### A. Phase-field approximation of diffuse crack topology

The diffuse crack surface area, in a regularized context, can be expressed in terms of crack surface density functional,  $\gamma(\phi, \nabla\phi)$  established by Miehe et al [35] in the localization band ( $\mathcal{X}$ ) in terms of phase field variable ( $\phi$ ) as,

$$\gamma(\phi, \nabla\phi) = \frac{1}{c_0} \left[ \frac{1}{l_0} \alpha(\phi) + l_0 |\nabla\phi|^2 \right] \quad (9)$$

$$\forall \quad c_0 = 4 \int_0^1 \sqrt{\alpha(z)} \, dz = 2 \quad (10)$$

The length scale parameter ( $l_0$ ) governs the width of the localization band ( $\mathcal{X}$ ) with  $c_0 > 0$  as a scaling factor. The evolution of crack is characterized by the crack geometric function  $\alpha(\phi) = \phi^2 \quad \forall \quad \alpha \in [0,1]$  as per the standard AT-2 model [66,67] resulting in the following crack surface density functional,

$$\gamma(\phi, \nabla\phi) = \left[ \frac{\phi^2}{2l_0} + \frac{l_0}{2} |\nabla\phi|^2 \right] \quad (11)$$

### B. Internal bulk energy and regularized dissipation functional

To account for the degradation of local strain energy density with crack growth, the internal bulk energy is expressed in terms of strain tensor ( $\boldsymbol{\epsilon} = \nabla^{sym}\mathbf{u}$ ) and a monotonically decreasing polynomial energy degradation function ( $g(\phi)$ ).

$$\begin{aligned} \Psi^b(\boldsymbol{\epsilon}, \phi) &= g(\phi) \cdot \Psi_0(\boldsymbol{\epsilon}) \\ \Psi_0(\boldsymbol{\epsilon}) &= \frac{1}{2} \boldsymbol{\epsilon} : \mathbf{D}_0 : \boldsymbol{\epsilon} = \frac{1}{2} \bar{\boldsymbol{\sigma}} : \mathbf{C}_0 : \bar{\boldsymbol{\sigma}} \end{aligned} \quad (12)$$

Here, effective stress tensor ( $\bar{\boldsymbol{\sigma}}$ ) for an isotropic elastic body can be defined in terms of elastic stiffness ( $\mathbf{D}_0$ ) and compliance ( $\mathbf{C}_0$ ) respectively as,

$$\begin{aligned} \bar{\boldsymbol{\sigma}} &= \mathbf{D}_0 : \boldsymbol{\epsilon}; \quad \boldsymbol{\epsilon} = \mathbf{C}_0 : \bar{\boldsymbol{\sigma}} \\ \mathbf{D}_0 &= \mathbf{C}_0^{-1} \\ \mathbf{D}_0 &= \lambda(\mathbf{I}_2 \otimes \mathbf{I}_2) + 2\mu\mathbf{I}_4 \end{aligned} \quad (13)$$

where,  $\mathbf{I}_j$  is  $j^{\text{th}}$  order Identity tensor with  $\lambda$  &  $\mu$  as Lamé's constants. The energy degradation function,  $g(\phi) \in [0,1]$ , chosen as a quadratic polynomial function [66,67], has a direct effect on the elastic stiffness of the material, ( $\mathbf{D} = g(\phi) \cdot \mathbf{D}_0$ ) and satisfies following properties [35],

$$g(\phi) = (1 - \phi^2) + \mathcal{K}_{stab} \quad g'(\phi) < 0, \quad g(0) = 1, g(1) = 0 \quad \text{and} \quad g'(1) = 0 \quad (14)$$

The stability parameter ( $\mathcal{K}_{stab}$ ) is taken as a small positive value, possibly of the order of  $10^{-7}$  to  $10^{-11}$ , to ensure the matrix conditioning while implementing numerical methods. This also ensures the existence of some residual material stiffness even after the diffuse crack has fully evolved. The evolution of crack in the regions of compressive and tensile stress fields is dissociated by considering either volumetric-deviatoric split or anisotropic split [34] of the local strain energy density. The former model by Amor et al [32] considers the following positive and negative components of local strain energy density,

$$\Psi_0(\boldsymbol{\epsilon}) \rightarrow \begin{cases} \Psi_0(\boldsymbol{\epsilon})^+ = \frac{1}{2} K_0 \langle tr(\boldsymbol{\epsilon}) \rangle^2 + \mu(\boldsymbol{\epsilon}_{dev} : \boldsymbol{\epsilon}_{dev}) \\ + \\ \Psi_0(\boldsymbol{\epsilon})^- = \frac{1}{2} K_0 \langle -tr(\boldsymbol{\epsilon}) \rangle^2 \end{cases} \quad (15)$$

The deviatoric component of the internal bulk energy contributes to the crack driving force for negative  $tr(\boldsymbol{\epsilon})$  in the aforementioned model, which partially inhibits the development of cracks under compression. Here, the Macaulay operator is defined as  $\langle V \rangle_{\pm} = \frac{V \pm |V|}{2}$  and  $K_0 = \lambda + 2\mu/3$  denotes the bulk modulus. The anisotropic model by Miehe et al [34] accounts for the evolution of crack under tension by splitting the strain tensor into positive and negative tensors for ' $\boldsymbol{\epsilon}_i$ ' and ' $\mathbf{n}_a$ ' as eigen values and eigen vectors respectively,

$$\Psi_0(\boldsymbol{\epsilon})^{\pm} = \frac{\lambda}{2} \langle tr(\boldsymbol{\epsilon}_i) \rangle_{\pm}^2 + \mu \langle tr^2(\boldsymbol{\epsilon}_i) \rangle_{\pm} \quad (16)$$

$$\boldsymbol{\epsilon} = \sum_{a=1}^3 \boldsymbol{\epsilon}_i \quad [\mathbf{n}_a \otimes \mathbf{n}_a] \quad (17)$$

The constitutive relations can be expressed as,

$$\boldsymbol{\sigma} = g(\phi) \cdot \bar{\boldsymbol{\sigma}} = \frac{\partial \Psi(\boldsymbol{\epsilon}, \phi)}{\partial \boldsymbol{\epsilon}} \quad (18)$$

$$\boldsymbol{\sigma} = \frac{\partial}{\partial \boldsymbol{\epsilon}} [g(\phi) \Psi_0^+ + \Psi_0^-] = g(\phi) \boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^- \quad (19)$$

To incorporate the cracking under tension and prevent interpenetration of crack faces under compression, only the positive part ( $\Psi_0^+$ ) is considered. A history field variable is introduced to ensure the damage irreversibility as maximum positive reference energy subjected to Karush-Kuhn-Tucker conditions [68] for both loading and unloading conditions,

$$\mathcal{H}^* = \begin{cases} \Psi_0^+(\epsilon) & \Psi_0^+ > \mathcal{H}_n \\ \mathcal{H}_n & \text{otherwise} \end{cases} \quad (20)$$

where  $\mathcal{H}_n$  is the history field variable calculated in the previous step. The variational of internal bulk energy (Eq. 12) is given by,

$$\delta\Psi^b(\epsilon, \phi) = \int_{\Omega} \left[ \frac{\partial\Psi^b}{\partial\epsilon} : \nabla^{sym}\delta\mathbf{u} + \frac{\partial\Psi^b}{\partial\phi} : \delta\phi \right] dV = \int_{\Omega} [\boldsymbol{\sigma} : \nabla^{sym}\delta\mathbf{u} + \frac{\partial\Psi^b}{\partial\phi} : \delta\phi] dV \quad (21)$$

The variational for the energy dissipated as the regularized crack topology (Eq. 11) evolves is expressed as,

$$\delta\Psi^{\Gamma}(\phi) = \int_{\Omega} G_c \left[ \frac{\phi}{l_0} \delta\phi + l_0 |\nabla\phi| \delta|\nabla\phi| \right] dV \quad (22)$$

### C. Governing equations

The variations of internal bulk energy (Eq. 12) and dissipation functional (Eq. 22) result in the following variational expressions (Eq. 7 & Eq. 8) in an updated context [41],

$$\delta\mathcal{W}_{int} = \int_{\Omega} [\boldsymbol{\sigma} : \nabla^{sym}\mathbf{u} + [g'(\phi). \mathcal{H}^*] : \delta\phi] + G_c \left[ \frac{\phi}{l_0} \delta\phi + l_0 |\nabla\phi| \delta|\nabla\phi| \right] dV \quad (23)$$

$$\delta\mathcal{W}_{ext} = \int_{\Omega} \bar{\mathbf{b}} \cdot \delta\mathbf{u} \, dV + \int_{\partial\Omega} \bar{\mathbf{t}} \cdot \delta\mathbf{u} \, dA \quad (24)$$

Thus, the potential energy functional minimization (Eq. 6) yields the following governing equations in weak form,

$$\int_{\Omega} [\nabla \cdot \boldsymbol{\sigma} + \bar{\mathbf{b}}] \delta\mathbf{u} \, dV + \int_{\partial\Omega_t} [\bar{\mathbf{t}} - \boldsymbol{\sigma} \cdot \mathbf{n}] \delta\mathbf{u} \, dA = 0 \quad (25)$$

$$\int_{\Omega} \left\{ g'(\phi). \mathcal{H}^* + G_c \left[ \frac{\phi}{l_0} - l_0 (\Delta\phi) \right] \right\} \delta\phi \, dV + \int_{\partial\Omega} G_c l_0 [\nabla\phi \cdot \hat{\mathbf{n}}_b] \delta\phi \, dA = 0 \quad (26)$$

here,  $\Delta$  represents the Laplacian operator and  $\hat{\mathbf{n}}_b$  is the outward unit normal vector on boundary  $\partial\mathcal{X}$ .

### D. Implementation of PFM with FEM

The nodal displacement field ( $\mathbf{u}$ ) and phase field ( $\phi$ ) can be expressed by discretizing the domain ( $\Omega$ ) in 2D linear quadrilateral and triangular elements as,

$$\mathbf{u} = \sum_{i=1}^m N_i^u u_i \quad \text{and} \quad \phi = \sum_{i=1}^m N_i^{\phi} \phi_i \quad (27)$$

where the element shape function matrix for the corresponding node  $i$  is defined as,

$$N_i^u = \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \quad \text{and} \quad N_i^{\phi} = N_i \quad (28)$$

The respective derivatives can be then defined as,

$$\boldsymbol{\epsilon} = \sum_{i=1}^m B_i^u u_i \quad \& \quad \nabla\phi = \sum_{i=1}^m B_i^{\phi} \phi_i \quad (29)$$

$$\text{for } B_i^u = \begin{bmatrix} N_{i,x} & 0 \\ 0 & N_{i,y} \\ N_{i,y} & N_{i,x} \end{bmatrix} \quad \& \quad B_i^{\phi} = \begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} \quad (30)$$

In the current research work, an alternating minimization scheme for the Newton-Raphson iterative method is used to obtain solutions for a system of nonlinear equations by updating the residual vector and tangent matrix at each iteration respectively,

$$\begin{bmatrix} K_s^{\phi} & 0 \\ 0 & K_s^u \end{bmatrix} \begin{Bmatrix} \phi_{s+1} \\ u_{s+1} \end{Bmatrix} = \begin{Bmatrix} R_s^{\phi} \\ R_s^u \end{Bmatrix} \quad (31)$$

where the corresponding residual vector and tangent matrix are expressed as follows,

$$K^{\phi} = \int_{\Omega} \left\{ \left[ \frac{G_c}{l_0} + 2\mathcal{H}^* \right] (N^{\phi})^T N^{\phi} + G_c l_0 (B^{\phi})^T B^{\phi} \right\} dV \quad (32)$$

$$K^u = \int_{\Omega} \{ g(\phi) (B^{\phi})^T \mathbb{D}_0 B^{\phi} \} dV \quad (33)$$

$$R^\phi = \int_{\Omega} \left\{ [g'(\phi)\mathcal{H}^* + \frac{G_c}{l_0}\phi](N^\phi)^T + G_c l_0 (B^\phi)^T \nabla \phi \right\} dV \quad (34)$$

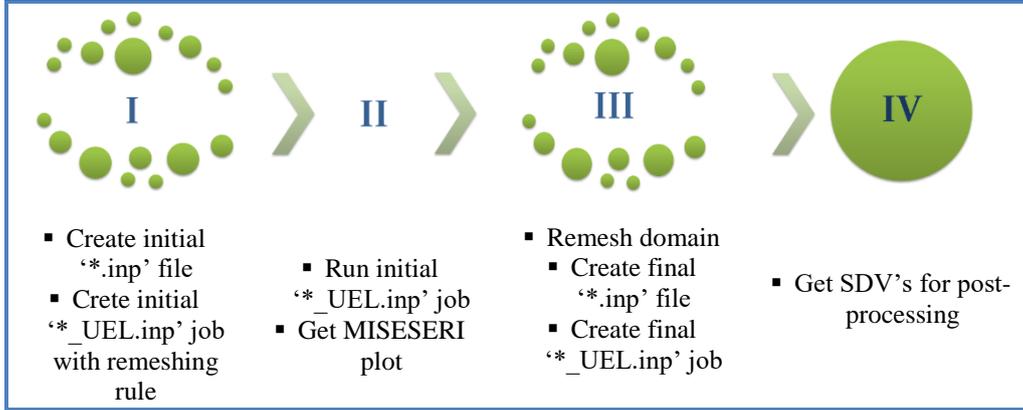
$$R^u = \int_{\Omega} g(\phi)(B^u)^T \bar{\sigma} dV - \int_{\Omega} (N^u)^T \bar{b} dV - \int_{\partial\Omega_t} (N^u)^T \bar{t} dA \quad (35)$$

### III. ADAPTIVE IMPLEMENTATION PROCEDURE IN ABAQUS

In the current work, we have adopted the layered approach [59] of the UEL subroutine utilizing linear triangular and linear quadrilateral user elements for prescribing element level calculations and UMAT for post-processing and feedback for error indicator MISESERI (Mises stress), which is a pre-requisite for establishing adaptive remeshing rule in Abaqus.

#### A. Error indicators

The commercial software Abaqus has built-in posteriori error indicator output variables such as MISESERI, ENDENERI (element energy density), and many more. These error indicators develop the groundwork for adaptive remeshing in Abaqus, which can be prompted by an adaptive remeshing rule. The posteriori error indicators reveal information about the regions of the analysis domain where discretization error potentially exists. These error indicators feature units that are identical to the base solution variable and can be requested as field output for the individual element. In the proposed mesh refinement algorithm, we have utilized the MISESERI error indicator variable as the basis of the remeshing rule for adaptive refinement at the desired locations. The MISESERI error indicator provides a discretization error estimate for the stress profile for the given boundary conditions.



**Figure 2: Block-wise overview of adaptive mesh refinement procedure**

There exists a difference in the order of assumed trial functions for displacement and stress approximations. The computations of the error indicators are based on the superconvergent patch recovery technique as reported in the literature [69]. The stress values at the integration points are computed by employing a higher-order trial function, whose order is the same as that of the displacement trial function. The energy norm distance between the values obtained by the smoothed estimation of the stress field and the non-smoothed estimation of the stress field serves as the value of the error estimate in the finite element solution. A detailed discussion on error estimation can be found in the literature [69–71].

#### B. Phase field implementation using UEL and UMAT subroutines

In Abaqus, the phase field implementation is accomplished by combining a UEL subroutine for integration point calculations and a UMAT subroutine for post-processing [59]. As a prerequisite, the UEL subroutine requires a `*_UEL.inp'` file which contains the essential inputs regarding the nodal repository, element connectivity for user-defined elements, and a set of elements `umatelem` which serves as a dummy layer for post-processing via UMAT subroutine. The initial input file `*.inp'` is generated by Abaqus itself for any analysis, however, for the phase field implementation, it is essential to define the user-defined elements, material & fracture parameters, and generate an element set for post-processing. All these essential data tweaks to the initial `*.inp'` file can be saved as `*_UEL.inp file'`, to be used in association with the UEL subroutine for phase field implementation. The UEL subroutine requires the user to provide the calculations for the tangent stiffness matrix (AMATRX) and force vector (RHS) for the element.

The material and fracture parameters specified in the `*_UEL.inp file'` can be incorporated into the subroutine as an argument (PROPS). The Abaqus solver interacts with the pre-defined array arguments of UEL to calculate and store results in the `*.odb'` file. The solution dependent state variable array (STATEV) in the

UMAT subroutine is utilized for post-processing the results. The detailed implementation of this approach is reported in the literature [58,59].

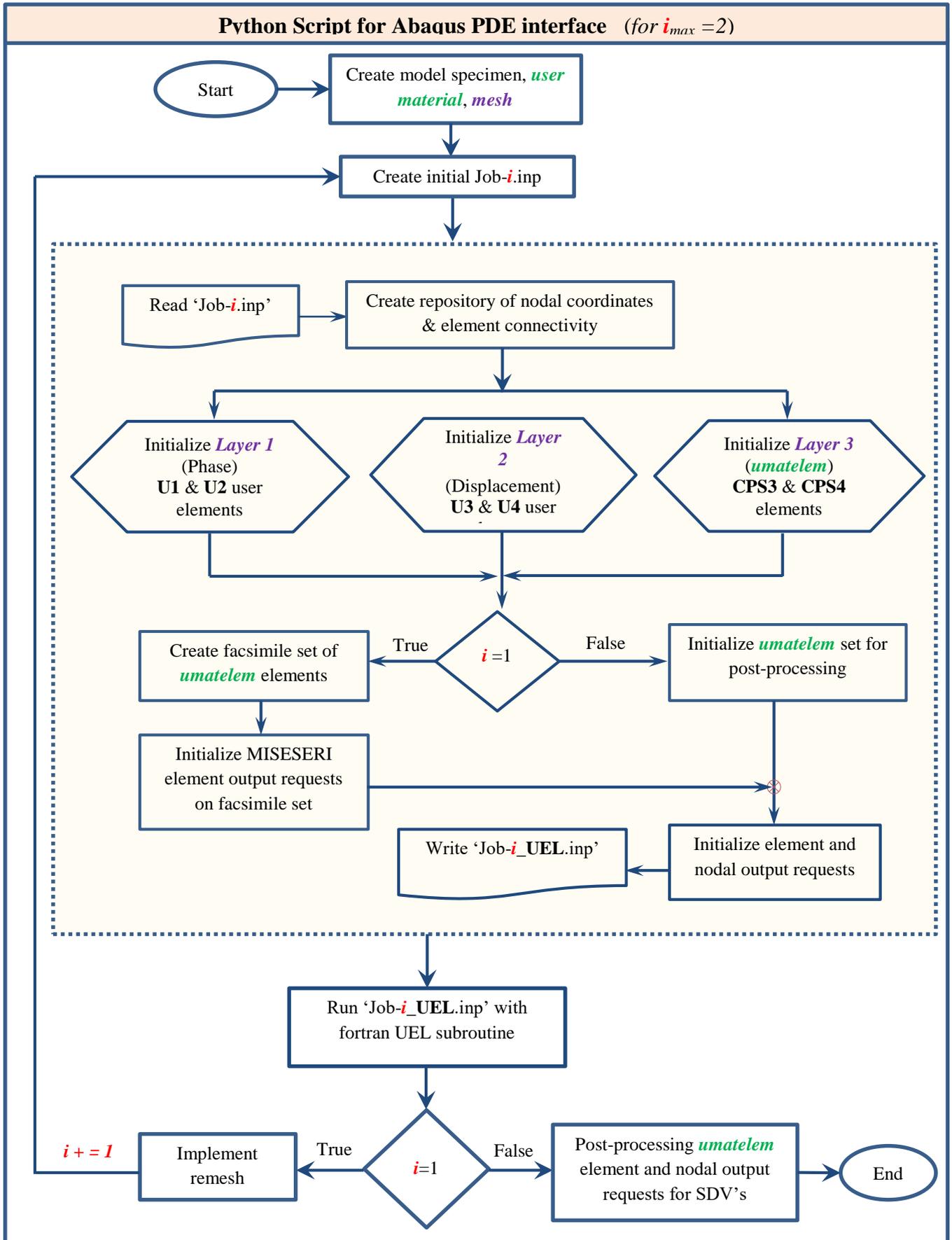


Figure 3: Flowchart of a Python script for proposed PFM in Abaqus

### C. Mesh refinement algorithm for phase field method

The proposed mesh refinement algorithm for the phase field method is implemented entirely through python scripting. The goal of employing the python scripting utility is to automate this entire implementation process in a single pass, which is disassembled block-by-block as illustrated in **Fig. 2** for illustration. The script contains appropriate functions/procedures for executing block-by-block operations simultaneously.

The process flowchart or the detailed operational plan depicting the algorithm for mesh refinement in PFM is shown in **Fig. 3**. The model geometry, user material data (with 16 dependent variables), section assignment, load step, boundary condition, amplitudes, etc. are defined initially through individual functions in the python script. The discretization of the specimen geometry defining element type and coarse mesh size is established next via a function in the python script. The adaptive mesh refinement consists of two simultaneous steps: first reporting an error indicator as field output (from step **I** to step **II** in **Fig. 2**) and then remeshing the designated regions so that the error estimate is within the tolerance limit or what is specified as `errorTarget` (refer **Block 1**) throughout the whole domain. This tolerance limit for the numerical examples, considered in the current work, is kept in the range of 1% - 5%, subjective to the type of problem. A remeshing rule in the part set '`All_elem`' as shown in **Block 1**. It should be noted that defining this part set '`All_elem`' is essential for establishing the remeshing rule in Abaqus. The set '`All_elem`' serves as the facsimile set with the same element connectivity as the set '`umatelem`' (**Block 2**). The integration point results obtained for the '`umatelem`' set are also available for this facsimile set, allowing for a remeshing rule to be established. The mesh parameters for executing mesh refinement can be specified as arguments to the `RemeshingRule` object (**Block 1**) as '`minElementSize`' and '`maxElementSize`', both being subjective to the length scale parameter considered for the problem ( $h/l_0$ ).

#### Block 1: A python function initializing MISESeri based remeshing rule with arguments

```
1 def create_remeshing_rule_assembly_instance (model_name, instance_name, step_name, maxSize,
2                                             minSize):
3     a = mdb.models[model_name].rootAssembly
4     set_name = instance_name + '.All_elem'
5     reg = a.sets[set_name]
6     m1 = mdb.models[model_name]
7     m1.RemeshingRule(name='RR: 1', stepName=step_name, region=reg,
8                     description="", outputFrequency=ALL_INCREMENTS,
9                     variables=('MISESeri', ), sizingMethod=UNIFORM_ERROR,
10                    errorTarget=1.0, specifyMinSize=True,
11                    specifyMaxSize=True, elementCountLimit=None,
12                    coarseningFactor=NOT_ALLOWED, refinementFactor=10,
13                    maxElementSize=maxSize, minElementSize=minSize)
```

#### Block 2: Initialization of facsimile set '`All_elem`' with element repository similar to '`umatelem`'

```
1 *Elset, elset=umatelem, generate
2 29609, 44412, 1
3 *Nset, nset=All_elem, generate
4 1, 14731, 1
5 *Elset, elset=All_elem, generate
6 29609, 44412, 1
7 ** FIELD OUTPUT: F-Output-1
8 **
9 *Output, field, time interval=0.001
10 *Element output, elset=Instance-1.All_elem, direction=YES
11 MISESeri, MISESAVG, S, EVOL
12 *Node Output, nset=REF_pt
13 RF, U
14 ** HISTORY OUTPUT: H-Output-1
15 **
16 *Element Output, elset=umatelem
17 SDV
18 *End Step
```

The python script contains a function that can read element connectivity and nodal repository from initial '`Job-1.inp`' and it returns '`Job-1_UEL.inp`' that has definitions for the layer system constituting user elements

U1 & U2 (phase), U3 & U4 (displacement) and post-processing element layer 3 documented. The material and fracture properties are passed to the subroutine via *'Job-1\_UEL.inp'* as user element properties in the input file [59]. This job is submitted for analysis to obtain the MISESERI values that are requested for the facsimile set *'All\_elem'* for each element as can be seen in **Block 3**. The MISESERI values indicate regions that require local refinement and then adaptive remeshing is implemented by calling a function in a python script that requires *'\*.odb'* file path and *'model\_name'* as arguments (**Block 4**). The adaptively refined mesh is registered as nodal and element repository by writing *'Job-2.inp'* which is used to create a new input file *'Job-2\_UEL.inp'* to be submitted along with the Fortran subroutine for post-processing SDV's for phase field results.

### Block 3: Element and nodal output requests for Abaqus job specified within the input file

```

1  ** OUTPUT REQUESTS
2  **
3  *Restart, write, overlay, frequency=0
4  **
5  ** FIELD OUTPUT: F-Output-1
6  **
7  *Output, field, time interval=0.001
8  *Element output, elset=Instance-1.All_elem, direction=YES
9  MISESERI, MISESAVG, S, EVOL
10 *Node Output, nset=REF_pt
11 RF, U
12 ** HISTORY OUTPUT: H-Output-1
13 **
14 *Element Output, elset=umatelem
15 SDV
16 *End Step

```

### Block 4: Python function to implement adaptive remeshing in Abaqus

```

1  def implement_remesh(odb_path, model_name):
2      session.viewports[myView].assemblyDisplay.setValues(mesh=ON)
3      o1 = odbAccess.openOdb(path=odb_path, readOnly=True) # o1 --> odb object
4      mdb.models[model_name].adaptiveRemesh(odb=o1)
6      a = mdb.models[model_name].rootAssembly # a --> assembly object
7      session.viewports[myView].setValues(displayObject=a)
8      session.viewports[myView].assemblyDisplay.setValues(mesh=ON)

```

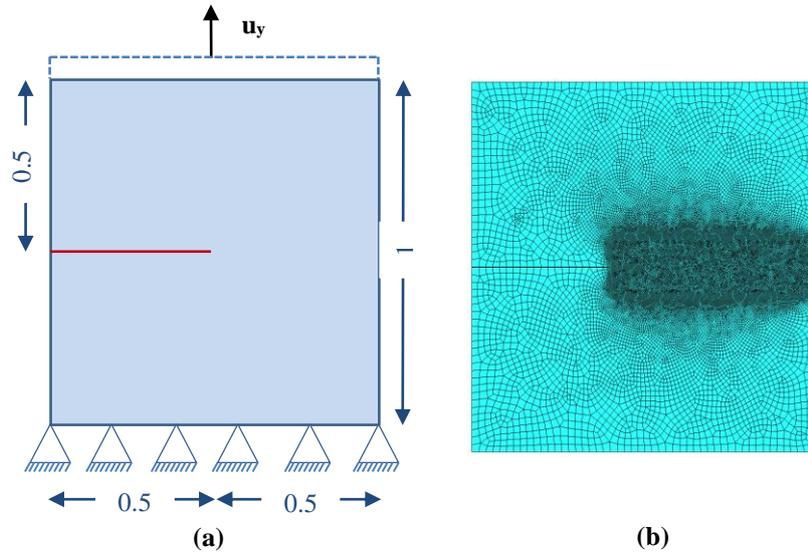
## IV. Numerical Results and Discussion

In this section, the validity and robustness of the proposed mesh refinement strategy for phase field method is validated through standard fracture problems. All the considered fracture problems are simulated with the standard phase field method as well as proposed mesh refinement based phase field method. The proposed-PFM results are compared with the standard PFM and available literature results. The computational efficiency of the proposed-PFM is shown through a pie chart for all the considered problems. In all the problems, the model is discretized using both triangular and quadrilateral elements in a non-uniform pattern. First, the standard problem of a single edge notch plate under mode I loading condition is considered for the study. Further, the L-shaped panel test is conducted to assess the capacity of the proposed approach to capture the crack nucleation, growth and failure in mixed mode loading conditions. Finally, the tensile failure of multi-hole specimens in absence of an initial crack is analyzed for a square panel with 5 holes. All the numerical simulations are executed on Dell Workstation with Intel(R) i9-10900 @ 2.80 GHz processor and 64GB RAM.

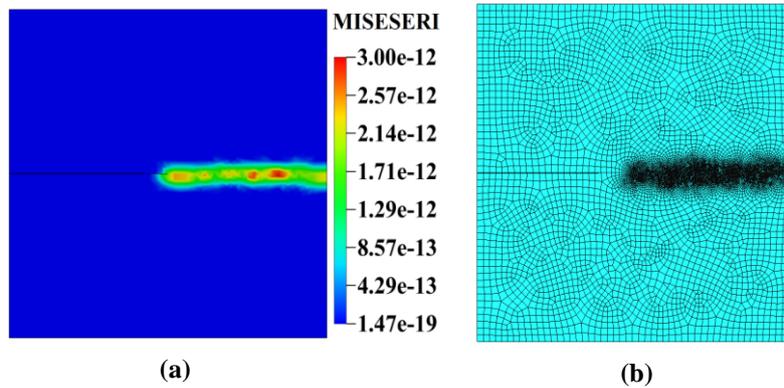
### A. Finite size edge crack plate under mode I loading

In the first example, a square plate with an edge crack of length 0.5 mm is considered for the simulations. The geometry and boundary conditions of the specimen are shown in **Fig. 4(a)**. The material properties of the specimen are taken from the literature [34], and are given as Young's modulus,  $E = 210$  GPa and Poisson's ratio,  $\nu = 0.3$ . The length scale parameter,  $l_0 = 0.0075$  mm and fracture energy,  $G_c = 2.7 \times 10^{-3}$  KN/mm are taken as fracture parameters.

At first, the problem is simulated using standard PFM, where the model is discretized with a total of 26282 linear quadrilateral and linear triangular elements with a global mesh size of 0.02 mm. Further, the mesh is refined with mesh size  $h = 0.003$  mm in the zone where the crack is expected to propagate as shown in **Fig. 4(b)**. This is done in accordance to maintain the suitable ratio of mesh size and length scale ( $h/l_0 \ll 1$ ) in the region. To study this problem, the anisotropic strain energy decomposition of Miehe et al [35] with staggered implementation [59] in Abaqus is considered. The loading is applied via displacement control strategy to determine the variation of load with respect to displacement. The displacement is applied in two steps. In the first step, the displacement  $u = 0.005$  mm is specified for 2000 increments at increment size  $\Delta u_1 = 10^{-4}$ . In the second step, the displacement is ramped to  $u = 0.01$  mm at  $\Delta u_2 = 10^{-5}$  for the remaining increments. This is achieved by defining the tabular amplitudes for each Abaqus step separately. The predicted crack path for standard PFM is illustrated by plotting the contours of the phase field variable ( $\phi$ ) as shown in **Fig. 6** for different displacement point.



**Figure 4: (a) Single edge crack specimen geometry, dimensions and boundary conditions for Mode I loading, (b) Discretized Abaqus model for standard PFM**

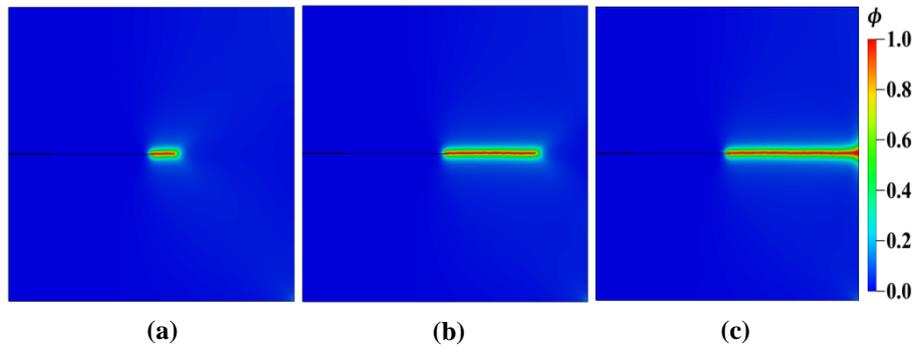


**Figure 5: Single edge crack specimen results for mode I loading; (a) MISESERRI plot prior to adaptive mesh refinement (b) Locally refined mesh based on MISESERRI plot with 13941 elements**

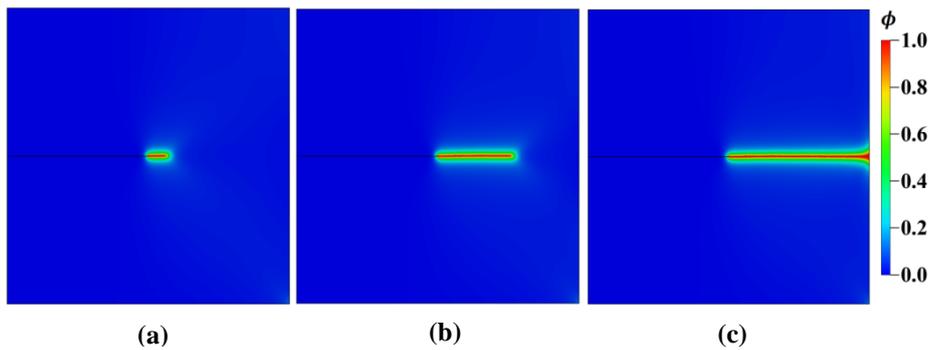
Further, the proposed PFM is implemented in Abaqus using a python script. The material and fracture parameters are taken the same as in standard PFM. The specimen is initially discretized with a global mesh size of 0.02 mm without local mesh refinement. A remeshing rule which governs the remesh region, global and local mesh size is established based on the MISESERRI error indicator in the Abaqus utility for the whole specimen using python scripting. To identify the MISESERRI values, a displacement control scheme is applied with increment size  $\Delta u_1 = 10^{-3}$  for 500 increments followed by  $\Delta u_2 = 5 \times 10^{-4}$  for the next 1000 increments. Based on the MISESERRI plot, the region as shown in **Fig. 5(a)** has been designated as the one that requires mesh refinement. The adaptive remesh is then incorporated with global mesh size  $\approx 0.02$  mm and local refined mesh size  $h = 0.001$  mm, comprising 13941 linear quadratic and triangular elements (**Fig. 5(b)**), which is quite lesser

than the standard PFM. The phase field results for proposed PFM are reported in **Fig. 7** which are similar to the standard PFM results shown in **Fig. 6**. Further, the load-displacement response for both methods i.e. standard PFM and proposed PFM is shown in **Fig. 8(a)** which are in good agreement with the literature [34,35].

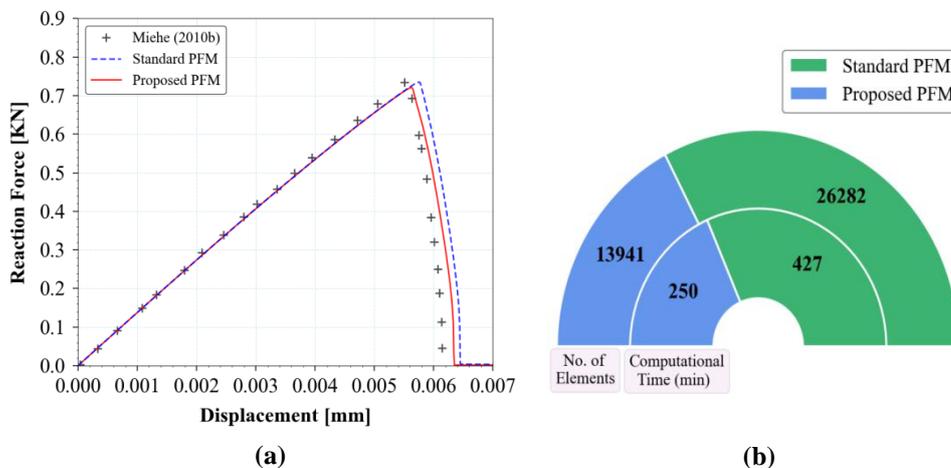
A comparison of computational efficiency for standard PFM and proposed PFM is shown in **Fig. 8(b)**. Compared to standard PFM, the global stiffness matrix size for proposed PFM is about half, allowing for less memory requirement. The CPU time required to solve the problem in standard PFM is 427 minutes, whereas it is 250 minutes for proposed PFM, indicating that the proposed PFM is more computationally efficient than the standard PFM.



**Figure 6: Standard PFM contours of phase field for mode I loading condition at  $l_0 = 0.0075$  mm for displacement; (a)  $u = 5.93 \times 10^{-3}$  mm (b)  $u = 6.29 \times 10^{-3}$  mm (c)  $u = 6.47 \times 10^{-3}$  mm**



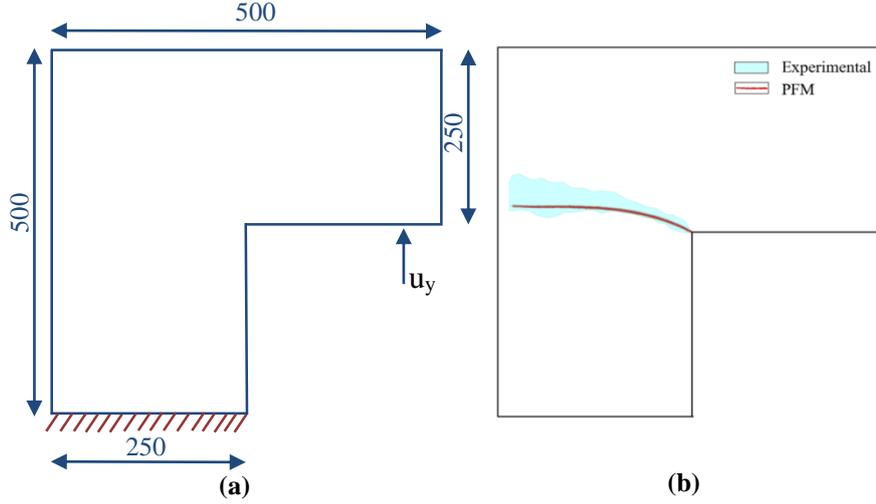
**Figure 7: Proposed PFM contours of phase field for mode I loading condition at  $l_0 = 0.0075$  mm for displacement; (a)  $u = 5.77 \times 10^{-3}$  mm (b)  $u = 6.1 \times 10^{-3}$  mm (c)  $u = 6.61 \times 10^{-3}$  mm**



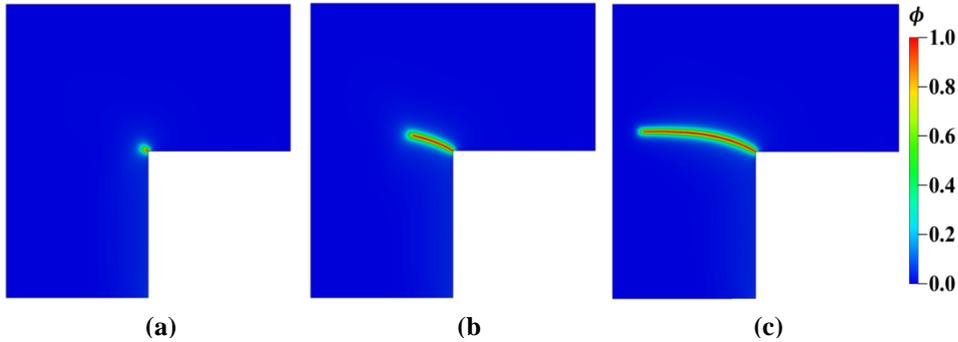
**Figure 8: (a) Structure response for single edge crack specimen under mode I loading, (b) comparison of number of elements and computational time between standard PFM & proposed**

## B. L-panel test

In this section, a mixed mode failure problem of an L-shaped panel is considered for the analysis. The geometry, dimensions and loading conditions are shown in **Fig. 9(a)**. The bottom edge is kept fixed while the incremental displacement is applied at a point on the right edge in the upward direction. The experimental analysis of this problem has been reported in the literature [72], and the experimentally observed crack path is depicted in **Fig. 9(b)** which is commonly considered as a validation benchmark for the crack path obtained from the numerical results. The material properties are taken as Young's modulus,  $E = 25.84$  GPa and Poisson's ratio,  $\nu = 0.18$ . The length scale parameter,  $l_0 = 3.125$  mm and fracture energy,  $G_c = 9.5 \times 10^{-5}$  KN/mm are taken as fracture parameters from the literature [73]. The thickness of the L-panel is taken as 100 mm.



**Figure 9: (a) Geometry and boundary conditions for an L-shaped panel, (b) Experimentally observed crack path domain [72]**

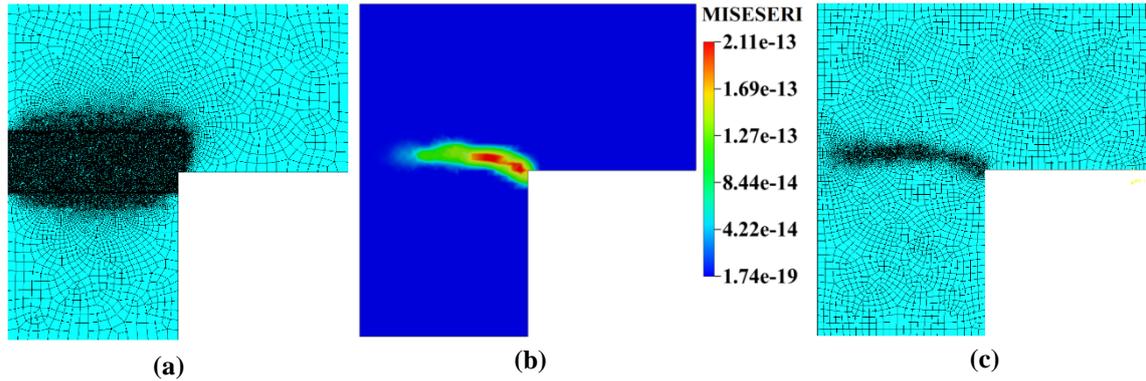


**Figure 10: Standard PFM contours of phase field for L-shaped panel at  $l_0 = 3.125$  mm for displacement; (a)  $u_{s1} = 0.276$  mm (b)  $u_{s2} = 0.332$  mm (c)  $u_{s3} = 0.544$  mm**

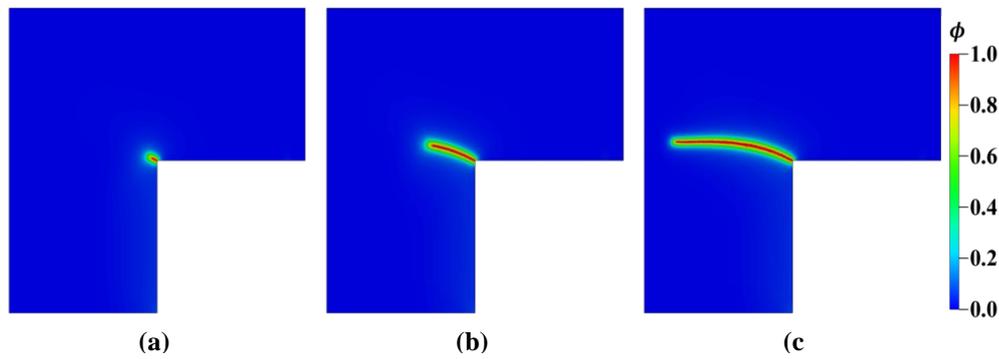
First, the simulation of an L-shaped panel is performed using standard PFM by discretizing the geometry with global mesh size of  $\approx 17$  mm. A local mesh refinement with element size 0.625 mm ( $h/l_0 = 0.2$ ) is adopted for the region depicted in **Fig. 11(a)**. The whole domain is discretized with non-uniform mesh consisting 55625 linear triangular and quadrilateral elements. A net displacement of 1 mm is applied in two steps at increment size of  $\Delta u_1 = 10^{-5}$  and  $\Delta u_2 = 10^{-6}$  for 1000 and 2000 increments, respectively to ensure that the structural response is well recorded. The structural response for the standard PFM is shown in **Fig. 13(a)**. The numerically predicted diffused crack pattern for standard PFM is illustrated in **Fig. 10** for different displacement points s1 to s2 marked in **Fig. 13(a)**.

Further, the simulation of L-shaped panel is performed using the proposed PFM. The material and fracture parameters are kept the same as in standard PFM. Initially, the global mesh size is taken as 12 mm for the whole model with no local mesh refinement. The region that requires local mesh refinement is obtained with the help of MISESeri values. For the initial 'Job-1\_UEL', a net displacement of 1 mm is applied in two steps with increment size of  $\Delta u_1 = 10^{-3}$  and  $\Delta u_2 = 10^{-4}$  for 1000 and 2000 increments, respectively, and the corresponding MISESeri plot is shown in **Fig. 11(b)**. The error-target value for the remeshing rule is kept at 3.5 on a 'UNIFORM\_ERROR' basis for the current example. The global mesh size and local mesh size ( $h$ ) are specified

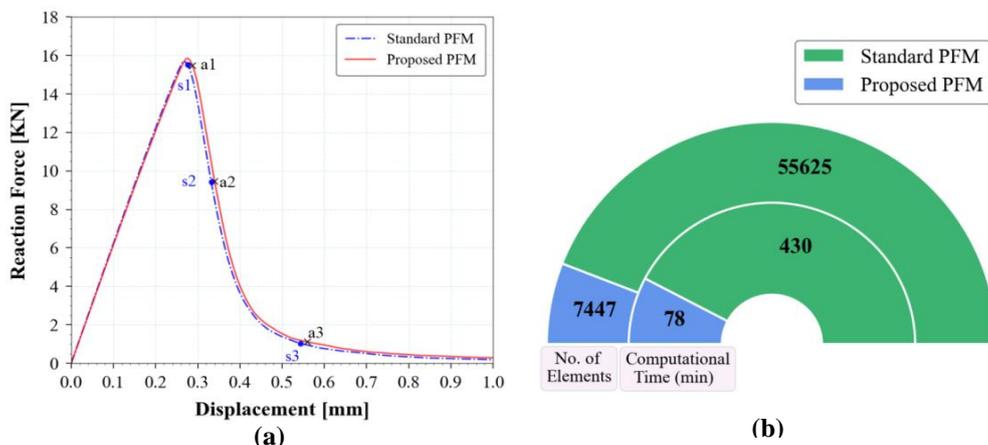
in the remeshing rule for the arguments '*maxElementSize*' and '*minElementSize*' (**Block 1**) as 15 mm and 0.625 mm respectively. The adaptively refined mesh for the L-shaped panel is depicted in **Fig. 11(c)**. The repository for nodal coordinates and element connectivity is read from '*Job-2.inp*' via python script and the input file '*Job-2\_UEL.inp*' required for final phase field results is developed. The incremental displacement of  $\Delta u_1 = 10^{-5}$  and  $\Delta u_2 = 10^{-6}$  for 1000 and 2000 increments, respectively, is applied to ensure the accurate structural response for proposed PFM. The phase field results for the proposed PFM are shown in **Fig. 12** corresponding to the points a1 to a3 marked in **Fig. 13(a)**. A good agreement between the standard PFM and proposed PFM regarding the crack path and structural response is observed which signifies that proposed adaptively refined PFM is able to successfully track the crack evolution mechanism for the L-shaped panel test. The comparison metrics for standard PFM and proposed PFM are presented in **Fig. 13(b)**. The computational time required for standard PFM is found to be 352 minutes more than the proposed PFM. Also, the memory requirement for later is found to be almost one-seventh of that of standard PFM.



**Figure 11:** (a) Standard PFM mesh for L-shaped panel with 55625 elements, (b) MISESERI plot for L-shaped panel, and (c) Adaptively refined mesh with 7447 elements



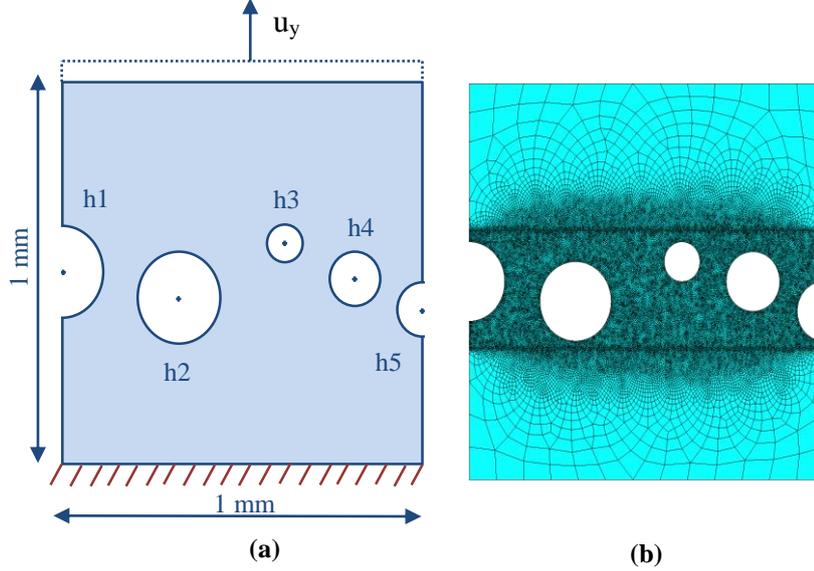
**Figure 12:** Proposed PFM contours of phase field for L-shaped panel at  $l_0 = 3.125$  mm for displacement; (a)  $u_{a1} = 0.288$  mm (b)  $u_{a2} = 0.34$  mm (c)  $u_{a3} = 0.56$  mm



**Figure 13:** (a) Load-displacement response corresponding for L-shaped panel, (b) Computational time and number of elements comparison between standard PFM and proposed PFM for the L-shaped panel

### C. Tensile failure of a square plate with five holes

The tensile failure of a square plate with five holes without any initial crack, is considered for the numerical analysis. The objective of studying this problem is to demonstrate the capacity of proposed mesh refinement strategy to capture the crack nucleation and growth in presence of multiple voids or defects, in the absence of any pre-existing crack. The geometry along with the applied boundary conditions is shown in **Fig. 14(a)**. The specimen consists of two semi-circular holes at the left and right edges and 3 circular holes of the different radii within the specimen [74]. The holes' geometric centers and radii are specified in **Table 1**. The volumetric-deviatoric energy split by Amor et al [32] is implemented in the Fortran subroutine for this problem.



**Figure 14: (a) Specimen geometry and boundary conditions for a square plate with 5 holes, (b) standard PFM discretized Abaqus model for square plate with 5 holes**

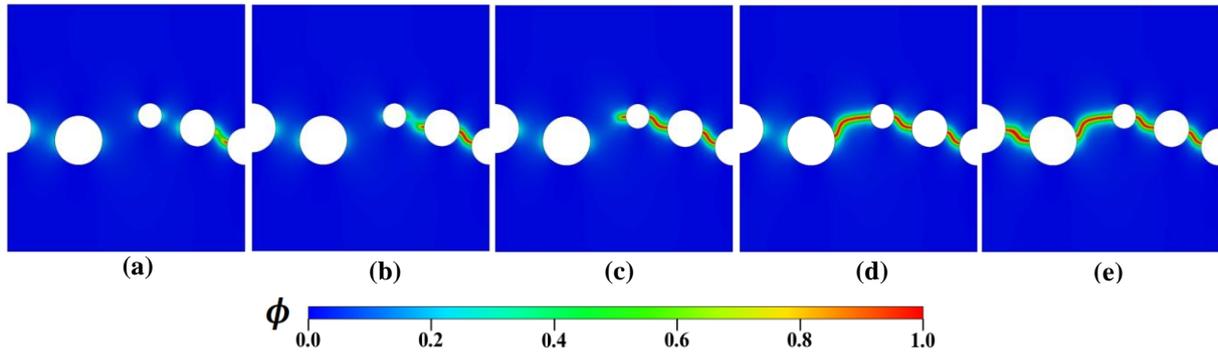
**Table 1: Coordinates of center and radius of holes for square plate with five holes**

Hole	Center coordinates	Radius (mm)
<b>h1</b>	(0, 0.5)	0.1
<b>h2</b>	(0.35, 0.45)	0.1
<b>h3</b>	(0.6, 0.55)	0.05
<b>h4</b>	(0.75, 0.5)	0.075
<b>h5</b>	(1, 0.45)	0.075

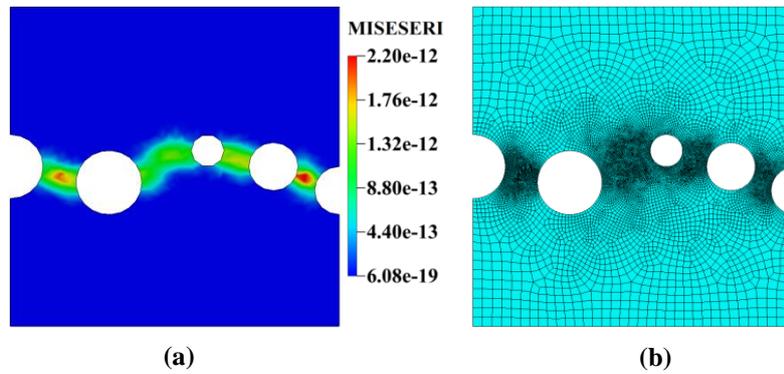
The problem is first simulated using the standard PFM method. The discretized specimen consisting of 45399 linear triangular and quadrilateral elements is shown in **Fig. 14(b)**. The total displacement  $u_y = 0.01$  mm is applied in two steps. In the first step, the increment size  $\Delta u_1 = 10^{-4}$  for 2000 increments, and for the second step, increment size  $\Delta u_2 = 5 \times 10^{-5}$  for 5000 increments is specified. The crack nucleation initiates near the hole **h5**, with a similar but much subdued nucleation in the vicinity of **h4**. Further, the crack grows along **h3** and **h2** with a modest increase in displacement. The crack growth mechanism observed between **h5** and **h4** is different from that of crack growth from **h4** to **h2**. The crack growth trajectory for standard PFM corresponding to the points marked in **Fig. 18(a)** is depicted in **Fig. 15** and is found to be in good agreement with the literature [74].

Further, the mesh refinement strategy implementation in PFM is executed via a two job process as explained in **Fig. 3**. The specimen is first discretized coarsely with a global mesh size of 0.02 mm. The '*Job-1\_UEL.inp*' is submitted along with the Fortran subroutine to obtain the MISESeri values. The MISESeri plot is depicted in **Fig. 16(a)**. The MISESeri plot represents the region requiring local mesh refinement which is implemented through the remeshing rule specified in **Block 4**. The adaptive remeshing enables a local mesh refinement of size  $h = 0.001$  mm and global mesh of size of 0.3 mm constituting total of 12944 elements as shown in **Fig. 16(b)**. The phase field results for the proposed PFM corresponding to the points marked in **Fig. 18(a)** are shown in **Fig. 17**. The crack growth pattern and mechanism for proposed PFM are found to be similar to that observed in standard PFM results and literature [74]. The load versus displacement plot for standard PFM and proposed PFM is reported in **Fig. 18(a)**. The number of elements and computational time is presented in **Fig. 18(b)**. The computational time for standard PFM and proposed PFM is found 1119 minutes and 365

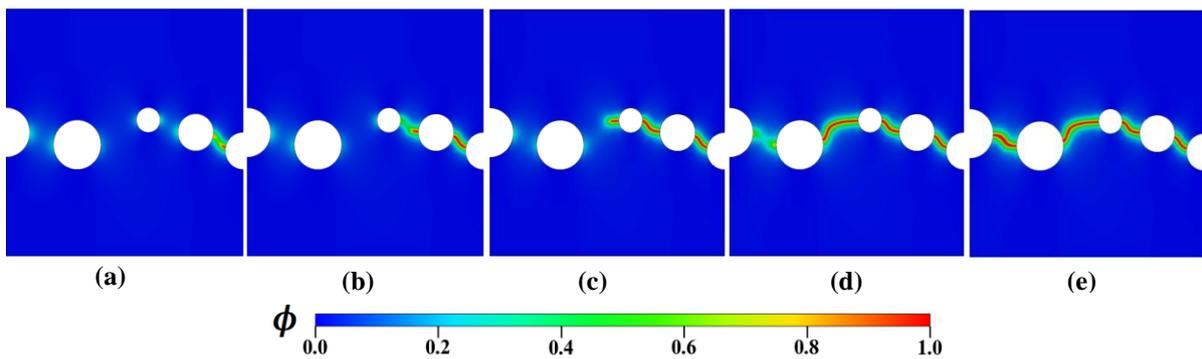
minutes respectively. Therefore, from the simulations, it can be concluded that the proposed PFM is computationally much more efficient for simulating such kind of problem.



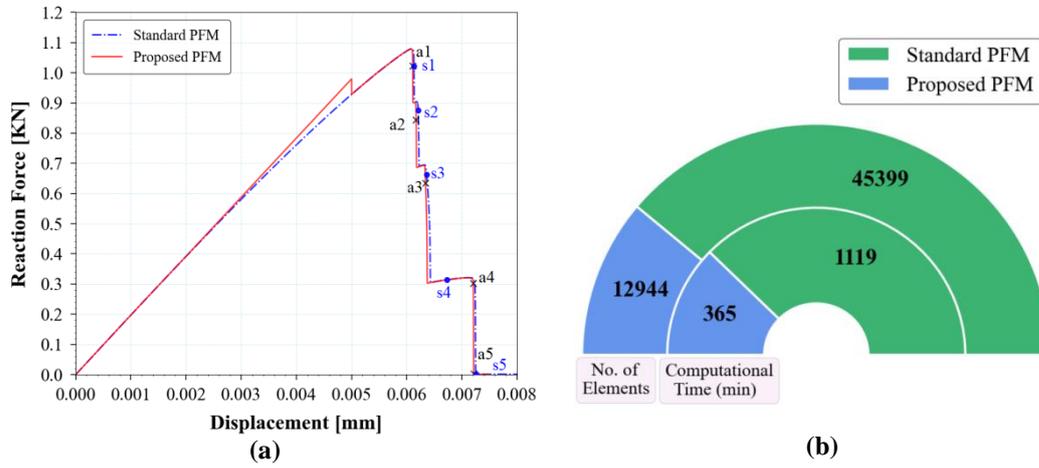
**Figure 15: Standard PFM contours of phase field for square plate specimen with 5 holes at  $l_0 = 0.008$  mm for displacement; (a)  $u_{s1} = 6.13 \times 10^{-3}$  mm (b)  $u_{s2} = 6.21 \times 10^{-3}$  mm (c)  $u_{s3} = 6.362 \times 10^{-3}$  mm (d)  $u_{s4} = 6.73 \times 10^{-3}$  mm (e)  $u_{s5} = 7.26 \times 10^{-3}$  mm**



**Figure 16: (a) MISESERI plot for a square plate specimen with 5 holes, (b) Adaptively refined mesh with 12944 elements for a square plate specimen with 5 holes**



**Figure 17: Proposed PFM contours of phase field for a square plate specimen with 5 holes at  $l_0 = 0.008$  mm for displacement; (a)  $u_{a1} = 6.108 \times 10^{-3}$  mm (b)  $u_{a2} = 6.173 \times 10^{-3}$  mm (c)  $u_{a3} = 6.343 \times 10^{-3}$  mm (d)  $u_{a4} = 7.206 \times 10^{-3}$  mm (e)  $u_{a5} = 7.223 \times 10^{-3}$  mm**



**Figure 18: (a) Load-displacement response for square plate specimen with 5 holes, (b) Computational time and number of elements comparison between standard PFM and proposed PFM**

## V. CONCLUSION

In this work, the implementation procedure of posteriori error indicator based adaptive mesh refinement algorithm in Abaqus is discussed for the phase field method. The proposed algorithm implementation in PFM is executed in Abaqus by a python script that features user-defined functions/procedures for the operational plan. The robustness and efficiency of the proposed approach are examined through several standard fracture problems. The conclusions drawn, based on the study, are given as:

- The proposed algorithm in PFM approach enables the use of Abaqus' built-in error indicators for adaptive mesh refinement instead of user-specified computations at the element level to mark regions requiring local mesh refinement.
- The proposed approach is able to successfully manifest the expected crack path through the Abaqus' built in MISES/ERI indicator.
- In the proposed PFM approach, the requirement to designate a high mesh density zone based on the prior information about the crack growth direction is entirely eliminated. Hence the proposed approach can be used for more generic problems.
- The proposed adaptive PFM is able to provide accurate results with a significant reduction in computational time as shown in **Table 2**.
- Although the proposed PFM performance and capability is positively demonstrated for 2D problems, extension to 3D problems is one of the potential future works to be considered.

**Table 2: Computational time-scale comparison between standard PFM and proposed PFM**

	Standard PFM (min)	Proposed PFM (min)	% Reduction in time
<b>Mode I</b>	427	250	41.45
<b>L-Panel</b>	430	78	81.86
<b>Square plate five holes</b>	1119	365	67.38

## REFERENCES

- [1] A.A. Griffith, VI. The phenomena of rupture and flow in solids, Philos. Trans. R. Soc. London. Ser. A, Contain. Pap. a Math. or Phys. Character. 221 (1921) 163–198.
- [2] G.R. Irwin, Analysis of stresses and strains near the end of a crack traversing a plate, (1957).
- [3] G.I. Barenblatt, The mathematical theory of equilibrium cracks in brittle fracture, Adv. Appl. Mech. 7 (1962) 55–129.
- [4] D.S. Dugdale, Yielding of steel sheets containing slits, J. Mech. Phys. Solids. 8 (1960) 100–104.
- [5] T. Hughes, The Finite Element Method. Linear Static and Dynamic Finite Element Analysis, 1987.
- [6] T. Belytschko, L. Gu, Y.Y. Lu, Fracture and crack growth by element free Galerkin methods, Model. Simul. Mater. Sci. Eng. (1994). <https://doi.org/10.1088/0965-0393/2/3A/007>.
- [7] H. Pathak, A. Singh, I.V. Singh, Numerical simulation of bi-material interfacial cracks using EFGM and XFEM, Int. J. Mech. Mater. Des. (2012). <https://doi.org/10.1007/s10999-011-9173-3>.
- [8] T.B. T. Belytschko, Elastic crack growth in finite elements with minimal remeshing, Int. J. Numer. Methods Eng. (1990).
- [9] S. Kumar, I. V. Singh, B.K. Mishra, T. Rabczuk, Modeling and simulation of kinked cracks by virtual node XFEM, Comput. Methods Appl. Mech. Eng. (2015). <https://doi.org/10.1016/j.cma.2014.10.019>.
- [10] S. Kumar, Y. Wang, L.H. Poh, B. Chen, Floating node method with domain-based interaction integral for generic 2D crack growths, Theor. Appl. Fract. Mech. (2018). <https://doi.org/10.1016/j.tafmec.2018.06.013>.
- [11] U. Singh, S. Kumar, B. Chen, Smoothed floating node method for modelling 2D arbitrary crack propagation problems, Theor. Appl. Fract. Mech. (2022). <https://doi.org/10.1016/j.tafmec.2021.103190>.

- [12] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Eng.* (2005). <https://doi.org/10.1016/j.cma.2004.10.008>.
- [13] A. Soni, A. Negi, S. Kumar, N. Kumar, An IGA based nonlocal gradient-enhanced damage model for failure analysis of cortical bone, *Eng. Fract. Mech.* (2021). <https://doi.org/10.1016/j.engfracmech.2021.107976>.
- [14] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, *Int. J. Numer. Methods Eng.* 45 (1999) 601–620.
- [15] T. Rabczuk, T. Belytschko, Cracking particles: A simplified meshfree method for arbitrary evolving cracks, *Int. J. Numer. Methods Eng.* (2004). <https://doi.org/10.1002/nme.1151>.
- [16] T. Rabczuk, T. Belytschko, A three-dimensional large deformation meshfree method for arbitrary evolving cracks, *Comput. Methods Appl. Mech. Eng.* (2007). <https://doi.org/10.1016/j.cma.2006.06.020>.
- [17] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, *J. Mech. Phys. Solids.* (2000).
- [18] S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, *J. Elast.* (2007). <https://doi.org/10.1007/s10659-007-9125-1>.
- [19] T. Ni, M. Zaccariotto, Q.Z. Zhu, U. Galvanetto, Coupling of FEM and ordinary state-based peridynamics for brittle failure analysis in 3D, *Mech. Adv. Mater. Struct.* (2021). <https://doi.org/10.1080/15376494.2019.1602237>.
- [20] L.M. KACHANOV, Time of the rupture process under creep conditions, *Izv Akad. Nank SSR Otd Tech Nauk.* 8 (1958) 26–31.
- [21] M. Jirásek, Mathematical analysis of strain localization, *Rev. Eur. Génie Civ.* 11 (2007) 977–991.
- [22] A. Karma, D.A. Kessler, H. Levine, Phase-field model of mode III dynamic fracture, *Phys. Rev. Lett.* 87 (2001) 45501.
- [23] V. Hakim, A. Karma, Laws of crack motion and phase-field models of fracture, *J. Mech. Phys. Solids.* 57 (2009) 342–368.
- [24] G.A. Francfort, J.J. Marigo, Revisiting brittle fracture as an energy minimization problem, *J. Mech. Phys. Solids.* 46 (1998) 1319–1342. [https://doi.org/10.1016/S0022-5096\(98\)00034-9](https://doi.org/10.1016/S0022-5096(98)00034-9).
- [25] D.B. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Commun. Pure Appl. Math.* (1989).
- [26] M. Buliga, Energy minimizing brittle crack propagation, *J. Elast.* 52 (1998) 201–238.
- [27] B. Bourdin, G.A. Francfort, J.J. Marigo, Numerical experiments in revisited brittle fracture, *J. Mech. Phys. Solids.* 48 (2000) 797–826. [https://doi.org/10.1016/S0022-5096\(99\)00028-9](https://doi.org/10.1016/S0022-5096(99)00028-9).
- [28] B. Bourdin, Numerical implementation of the variational formulation for quasi-static brittle fracture, *Interfaces Free Boundaries.* 9 (2007) 411–430.
- [29] B. Bourdin, G.A. Francfort, J.-J. Marigo, The variational approach to fracture, *J. Elast.* 91 (2008) 5–148.
- [30] Y. Kriaa, H. Hentati, B. Zouari, Applying the phase-field approach for brittle fracture prediction: Numerical implementation and experimental validation, *Mech. Adv. Mater. Struct.* (2020). <https://doi.org/10.1080/15376494.2020.1795957>.
- [31] J. Dujc, B. Brank, Combining coupled, staggered and uncoupled solution methods for phase-field-based fracture analysis, *Mech. Adv. Mater. Struct.* (2021). <https://doi.org/10.1080/15376494.2021.1976888>.
- [32] H. Amor, J.-J. Marigo, C. Maurini, Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments, *J. Mech. Phys. Solids.* 57 (2009) 1209–1229.
- [33] G. Lancioni, G. Royer-Carfagni, The variational approach to fracture mechanics. a practical application to the french panthéon in Paris, *J. Elast.* 95 (2009) 1–30. <https://doi.org/10.1007/s10659-009-9189-1>.
- [34] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, *Comput. Methods Appl. Mech. Eng.* 199 (2010) 2765–2778.
- [35] C. Miehe, F. Welschinger, M. Hofacker, Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations, *Int. J. Numer. Methods Eng.* 83 (2010) 1273–1311.
- [36] J.-Y. Wu, M. Cervera, A novel positive/negative projection in energy norm for the damage modeling of quasi-brittle solids, *Int. J. Solids Struct.* 139 (2018) 250–269.
- [37] J.-Y. Wu, V.P. Nguyen, A length scale insensitive phase-field damage model for brittle fracture, *J. Mech. Phys. Solids.* 119 (2018) 20–42.
- [38] M. Ambati, T. Gerasimov, L. De Lorenzis, A review on phase-field models of brittle fracture and a new fast hybrid formulation, *Comput. Mech.* 55 (2015) 383–405.
- [39] J.-Y. Wu, V.P. Nguyen, C.T. Nguyen, D. Sutula, S. Sinaie, S.P.A. Bordas, Phase-field modeling of fracture, *Adv. Appl. Mech.* 53 (2020) 1–183.
- [40] Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda, A simple and robust Abaqus implementation of the phase field fracture method, *Appl. Eng. Sci.* 6 (2021) 100050.
- [41] E. Martínez-Pañeda, A. Golahmar, C.F. Niordson, A phase field formulation for hydrogen assisted cracking, *Comput. Methods Appl. Mech. Eng.* 342 (2018) 742–761.
- [42] M.J. Borden, T.J.R. Hughes, C.M. Landis, C. V Verhoosel, A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework, *Comput. Methods Appl. Mech. Eng.* 273 (2014) 100–118.
- [43] M. Paggi, M. Corrado, J. Reinoso, Fracture of solar-grade anisotropic polycrystalline Silicon: A combined phase field–cohesive zone model approach, *Comput. Methods Appl. Mech. Eng.* 330 (2018) 123–148.
- [44] M. Klinsmann, D. Rosato, M. Kamlah, R.M. McMeeking, An assessment of the phase field formulation for crack growth, *Comput. Methods Appl. Mech. Eng.* 294 (2015) 313–330.
- [45] T. Heister, M.F. Wheeler, T. Wick, A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach, *Comput. Methods Appl. Mech. Eng.* 290 (2015) 466–495.
- [46] A. Gupta, U.M. Krishnan, T.K. Mandal, R. Chowdhury, V.P. Nguyen, An adaptive mesh refinement algorithm for phase-field fracture models: Application to brittle, cohesive, and dynamic fracture, *Comput. Methods Appl. Mech. Eng.* 399 (2022) 115347. <https://doi.org/10.1016/j.cma.2022.115347>.
- [47] M. Artina, M. Fornasier, S. Micheletti, S. Perotto, Anisotropic mesh adaptation for crack detection in brittle materials, *SIAM J. Sci. Comput.* 37 (2015) B633–B659.
- [48] R.U. Patil, B.K. Mishra, I. Singh, An adaptive multiscale phase field method for brittle fracture, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 254–288.
- [49] A.L.N. Pramod, R.K. Annabattula, E.T. Ooi, C. Song, S. Natarajan, others, Adaptive phase-field modeling of brittle fracture using the scaled boundary finite element method, *Comput. Methods Appl. Mech. Eng.* 355 (2019) 284–307.
- [50] C. Jansari, K. Kannan, R.K. Annabattula, S. Natarajan, others, Adaptive phase field method for quasi-static brittle fracture using a recovery based error indicator and quadtree decomposition, *Eng. Fract. Mech.* 220 (2019) 106599.
- [51] F. Peng, H. Liu, S. Li, X. Cui, Quadtree-polygonal smoothed finite element method for adaptive brittle fracture problems, *Eng. Anal. Bound. Elem.* 134 (2022) 491–509.

- [52] K. Mang, M. Walloth, T. Wick, W. Wollner, Mesh adaptivity for quasi-static phase-field fractures based on a residual-type a posteriori error estimator, *GAMM-Mitteilungen*. 43 (2020) e202000003.
- [53] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, *Theor. Appl. Fract. Mech.* (2020). <https://doi.org/10.1016/j.tafmec.2019.102447>.
- [54] P. Farrell, C. Maurini, Linear and nonlinear solvers for variational phase-field models of brittle fracture, *Int. J. Numer. Methods Eng.* 109 (2017) 648–667.
- [55] S. May, J. Vignollet, R. De Borst, A numerical assessment of phase-field models for brittle and cohesive fracture:  $\Gamma$ -convergence and stress oscillations, *Eur. J. Mech.* 52 (2015) 72–84.
- [56] D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandié, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, *Nucl. Eng. Des.* (2009). <https://doi.org/10.1016/j.nucengdes.2009.05.021>.
- [57] J.Y. Wu, Y. Huang, V.P. Nguyen, On the BFGS monolithic algorithm for the unified phase field damage theory, *Comput. Methods Appl. Mech. Eng.* (2020). <https://doi.org/10.1016/j.cma.2019.112704>.
- [58] M.A. Msekh, J.M. Sargado, M. Jamshidian, P.M. Areias, T. Rabczuk, Abaqus implementation of phase-field model for brittle fracture, *Comput. Mater. Sci.* 96 (2015) 472–484.
- [59] G. Molnár, A. Gravouil, 2D and 3D Abaqus implementation of a robust staggered phase-field solution for modeling brittle fracture, *Finite Elem. Anal. Des.* 130 (2017) 27–38.
- [60] J.-Y. Wu, Y. Huang, Comprehensive implementations of phase-field damage models in Abaqus, *Theor. Appl. Fract. Mech.* 106 (2020) 102440.
- [61] G. Liu, Q. Li, M.A. Msekh, Z. Zuo, Abaqus implementation of monolithic and staggered schemes for quasi-static and dynamic fracture phase-field model, *Comput. Mater. Sci.* 121 (2016) 35–47.
- [62] U. Pillai, Y. Heider, B. Markert, A diffusive dynamic brittle fracture model for heterogeneous solids and porous materials with implementation using a user-element subroutine, *Comput. Mater. Sci.* 153 (2018) 36–47.
- [63] H. Jeong, S. Signetti, T.-S. Han, S. Ryu, Phase field modeling of crack propagation under combined shear and tensile loading with hybrid formulation, *Comput. Mater. Sci.* 155 (2018) 483–492.
- [64] S.S.K. Reddy, R. Amirham, J.N. Reddy, Modeling fracture in brittle materials with inertia effects using the phase field method, *Mech. Adv. Mater. Struct.* (2021). <https://doi.org/10.1080/15376494.2021.2010289>.
- [65] J.Y. Wu, A unified phase-field theory for the mechanics of damage and quasi-brittle failure, *J. Mech. Phys. Solids.* (2017). <https://doi.org/10.1016/j.jmps.2017.03.015>.
- [66] L. Ambrosio, V.M. Tortorelli, Approximation of functional depending on jumps by elliptic functional via  $t$ -convergence, *Commun. Pure Appl. Math.* 43 (1990) 999–1036.
- [67] B. Bourdin, G.A. Francfort, J.-J. Marigo, Numerical experiments in revisited brittle fracture, *J. Mech. Phys. Solids.* 48 (2000) 797–826.
- [68] N. Singh, C. V Verhoosel, R. De Borst, E.H. Van Brummelen, A fracture-controlled path-following technique for phase-field modeling of brittle fracture, *Finite Elem. Anal. Des.* 113 (2016) 14–29.
- [69] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique, *Int. J. Numer. Methods Eng.* 33 (1992) 1331–1364.
- [70] O.C. Zienkiewicz, J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, *Int. J. Numer. Methods Eng.* 24 (1987) 337–357.
- [71] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity, *Int. J. Numer. Methods Eng.* 33 (1992) 1365–1382.
- [72] B. Winkler, G. Hofstetter, G. Niederwanger, Experimental verification of a constitutive model for concrete cracking, *Proc. Inst. Mech. Eng. Part L J. Mater. Des. Appl.* 215 (2001) 75–86. <https://doi.org/10.1177/146442070121500202>.
- [73] O. Lampron, D. Theriault, M. Lévesque, An efficient and robust monolithic approach to phase-field quasi-static brittle fracture using a modified Newton method, *Comput. Methods Appl. Mech. Eng.* 386 (2021). <https://doi.org/10.1016/j.cma.2021.114091>.
- [74] J. Han, Y. Shintaku, S. Moriguchi, K. Terada, A transition scheme from diffusive to discrete crack topologies under finite strain during the course of a staggered iterative procedure, (n.d.).