ABAQUS implementation of the phase field fracture method

Emilio Martínez-Pañeda[a,*], Alireza Golahmar[b]

[a] *Department of Engineering, Cambridge University, CB2 1PZ Cambridge, UK*
[b] *Department of Mechanical Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

Documentation that accompanies the user element (UEL) subroutine for implementing the phase field model for fracture in Abaqus. Several Fortran files are provided, each associated with a particular choice of element or time integration scheme, along with the corresponding input files (see Appendix B). If using this code for research or industrial purposes, please cite:

E. Martínez-Pañeda, A. Golahmar, C.F. Niordson. A phase field formulation for hydrogen assisted cracking. Computer Methods in Applied Mechanics and Engineering 342: 742-761 (2018)

Keywords:
ABAQUS, Phase field, Fracture, Crack growth, Finite element analysis

1. The phase field method for fracture

The phase field model for fracture builds upon the pioneering thermodynamic framework established by Griffith, where crack growth will take place if a critical energy release rate is attained. Frankfort and Marigo [1] were the first to embed Griffith's approach into variational formulations by including in the total potential energy the surface energy dissipated by the formation of a crack. Bourdin et al. [2, 3] regularized the discrete crack

*Corresponding author. Tel: +44 1223 7 48525.
Email address: mail@empaneda.com (Emilio Martínez-Pañeda)

topology by means of a scalar damage variable and a diffuse crack representation. This variable is termed as the phase field, or phase field order parameter. Important contributions to the model have also been made by Miehe and co-workers [4, 5]. Due to its robustness, the phase field fracture model enjoys great popularity and has not only been successfully applied to model brittle fracture but also to ductile damage [6, 7], hydraulic fracturing [8, 9], composites delamination [10, 11], and hydrogen assisted cracking [12], to name a few. We provide an efficient and robust implementation of the phase field method in the commercial finite element package Abaqus, enabling to model interactions and branching of cracks of arbitrary topological complexity. Readers familiar with the theoretical and numerical foundations of the phase field method can jump to Section 2.3 to read about the usage of the files provided.

1.1. Representation of crack surface density function

Consider a 1D problem consisting of an infinitely expanded bar of constant cross sectional area Γ being aligned along the x-axis and occupying the solid domain Ω (see Fig. 1a). Assume that the bar is fully cut by a crack at the axial position $x = 0$, where Γ represents the fully broken crack surface. The *sharp crack topology* (see Fig. 1b) can be described by an auxiliary field variable $\phi(x) \in [0, 1]$ with (Dirac delta function)

$$\phi(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \tag{1}$$

Here, $\phi(x)$ is denoted as the *crack phase field*, with $\phi = 0$ characterizing a completely intact state and $\phi = 1$ a fully broken state of the material. As such, a suitable approach, grounded on continuum damage mechanics [1], is to smear out the crack over the axial domain of the bar. Thus, the non-smooth crack phase field (1) can be approximated by the exponential function (2),

$$\phi(x) = \mathrm{e}^{-\frac{|x|}{\ell}} \tag{2}$$

that represents a *regularized* or *diffusive crack topology* as illustrated in Fig. 1c.

[1] The crack itself is not a discrete phenomenon but initiates with micro-cracks and nano voids.

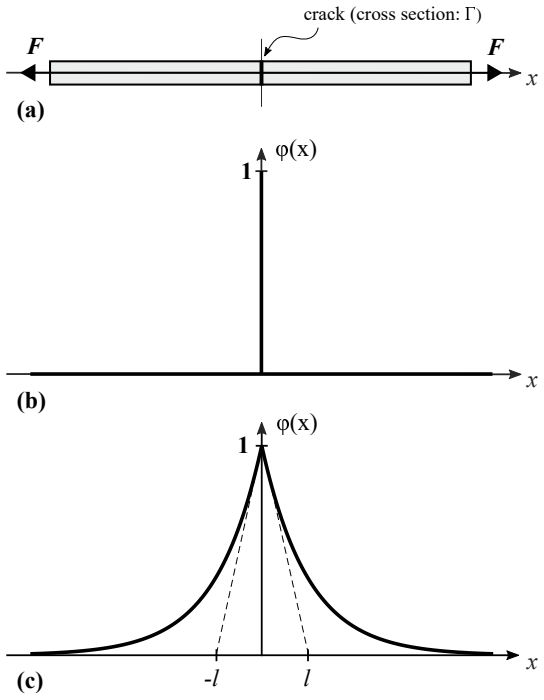

Figure 1: Sharp and diffusive crack modeling. (a) 1D bar cut by a crack with cross section Γ. (b) Sharp crack at $x = 0$. (c) Diffusive crack at $x = 0$ modeled with length scale ℓ.

The length-scale parameter ℓ governs the amount of diffusion of the crack, and for $\ell \to 0$ the function (2) equates (1), meaning that the sharp crack case is approached. It should be emphasized that ℓ does not represent the actual width of the diffusive crack, since the crack is smoothed out over the entire domain. The exponential function (2) is actually never zero but decays monotonically to zero as we move away from the crack location. Thus,

$$\begin{gathered} \phi(0) = 1 \\ \phi(x) \to 0 \quad \text{as} \quad x \to \pm\infty \end{gathered} \tag{3}$$

Taking derivatives of (2) gives

$$\begin{aligned} \phi^{'}(x) &= -\frac{\operatorname{sgn}(x)}{\ell} \mathrm{e}^{-\frac{|x|}{\ell}} \\ \phi^{''}(x) &= \left[\frac{\operatorname{sgn}(x)}{\ell}\right]^2 \mathrm{e}^{-\frac{|x|}{\ell}} = \frac{1}{\ell^2} \mathrm{e}^{-\frac{|x|}{\ell}} = \frac{1}{\ell^2}\, \phi(x) \end{aligned} \tag{4}$$

implying that (2) is the solution of the following ordinary differential equation for $x \neq 0$

$$\phi^{''}(x) - \frac{1}{\ell^2}\,\phi(x) = 0 \quad \text{in} \quad \Omega \tag{5}$$

subject to Dririchlet-type boundary conditions (3). Thus, the differential equation (5) is the Euler-Lagrange equation of the variational problem

$$\phi = \arg\left\{ \inf_{\phi \in W} I(\phi) \right\} \tag{6}$$

in which W is the set of functions satisfying (3). The associated quadratic functional is then given by

$$I(\phi) = \int_{-\infty}^{\infty} \frac{1}{2}\left[\phi^2 + \ell^2(\phi^{'})^2\right]\,\mathrm{d}x \tag{7}$$

The functional (7) is simply constructed by integrating a Galerkin-type weak formulation (8) of the differential equation (5)

$$\partial I(\phi) = \int_{-\infty}^{\infty} \frac{1}{2}\left[\upsilon\,\phi + \ell^2 \upsilon^{'}\phi^{'}\right]\,\mathrm{d}x = 0 \tag{8}$$

where υ represents a test function. Plugging (2) into the functional (7) yields the identification

$$I(\phi) = \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{2|x|}{\ell}}\,\mathrm{d}x = \ell \tag{9}$$

Now, considering the integration over the volume domain $\mathrm{d}V = \Gamma \mathrm{d}x$,

$$I(\phi = \mathrm{e}^{-\frac{|x|}{\ell}}) = \ell\Gamma \tag{10}$$

Thus, the crack surface Γ is related to the crack length scale parameter ℓ. Consequently, the crack surface density is introduced by means of the regularized crack functional as

$$\Gamma_\ell(\phi) = \frac{1}{\ell} I(\phi) = \int_\Omega \frac{1}{2\ell}\left[\phi^2 + \ell^2(\phi^{'})^2\right]\,\mathrm{d}V = \int_\Omega \gamma(\phi,\,\phi^{'})\,\mathrm{d}V \tag{11}$$

where $\gamma(\phi,\,\phi^{'})$ is the crack surface density function in 1D. Similarly, in higher dimensions it can be expressed as

$$\gamma(\phi, \nabla\phi) = \frac{1}{2\ell}\phi^2 + \frac{\ell}{2}|\nabla\phi|^2 \tag{12}$$

1.2. Strain energy degradation

In order to couple the fracture phase field with the deformation problem, the total potential energy functional of a solid body takes the following form

$$\Psi(\mathbf{u}, \phi) = \Psi^b(\mathbf{u}, \phi) + \Psi^s(\phi) \tag{13}$$

where the former term refers to the stored bulk energy while the latter term refers to the surface energy associated with the formation of a crack. Let $\Omega \subset \mathbb{R}^\delta$ be the reference configuration of a material body with dimension $\delta \in [1, 2, 3]$ in space and $\delta \subset \mathbb{R}^{\delta-1}$ its surface as illustrated in Fig. 2. The crack and the displacement field is studied in the range $\mathrm{T} \subset \mathbb{R}$ of time. Consequently, the time-dependent crack phase field is introduced as

$$\phi : \begin{cases} \Omega \times \mathrm{T} \ \rightarrow \ [0, 1] \\ (x, t) \ \rightarrow \ \phi(x, t) \end{cases} \tag{14}$$

and the displacement field as

$$\mathbf{u} : \begin{cases} \Omega \times \mathrm{T} \ \rightarrow \ \mathbb{R}^\delta \\ (x, t) \ \rightarrow \ \mathbf{u}(x, t) \end{cases} \tag{15}$$

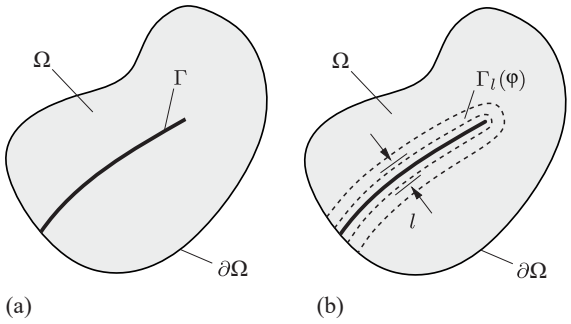

Figure 2: Sharp and diffusive crack topology. Adapted from [5]. (a) The sharp crack surface Γ embedded into the solid Ω and (b) the diffusive crack surface $\Gamma_\ell(\phi)$ is a functional of the crack phase field ϕ.

1.2.1. Isotropic degradation of the stored bulk energy

In the total potential energy functional (13), the stored bulk energy can be written as

$$\Psi^b(\mathbf{u},\phi) = \int_\Omega \psi\left(\boldsymbol{\epsilon}(\mathbf{u}),\phi\right)\, \mathrm{d}V \tag{16}$$

that depends on the displacement $\mathbf{u}$ and the fracture phase field ϕ. The *energy storage function* ψ describes the stored bulk energy of the solid per unit volume

$$\psi\left(\boldsymbol{\epsilon},\phi\right) = g(\phi)\psi_0(\boldsymbol{\epsilon}) \tag{17}$$

where ψ_0 is the elastic strain energy density. Assuming the standard linear theory of elasticity for the unbroken isotropic solid, the elastic strain energy density takes the form

$$\psi_0(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon}^T\mathbf{C_0}\boldsymbol{\epsilon} \tag{18}$$

with $\mathbf{C}_0$ being the linear elastic stiffness matrix in Voigt notation. Linear elasticity is assumed here for simplicity but the extension to J2 plasticity is straightforward and can be provided upon request. Additionally, one should note that a linear elastic description of crack tip stresses is closer to the predictions of implicitly multi-scale plasticity formulations (e.g., strain gradient plasticity) than conventional plasticity [13, 14]. Considering the plane strain condition in 2D

$$\mathbf{C}_0 = \tfrac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{19}$$

with E and ν being the Young's modulus and Poisson's ratio, respectively. Assuming small strains, the standard strain tensor $\boldsymbol{\epsilon}$ can be defined as

$$\boldsymbol{\epsilon} = \frac{1}{2}\left[\nabla^T\mathbf{u} + \nabla\mathbf{u}\right] \tag{20}$$

where $\mathbf{u}$ is the vector of displacements and $\nabla\mathbf{u}$ its gradient. The monotonically decreasing function $g(\phi)$ describes the degradation of the stored bulk energy due to damage evolution by,

$$g(0) = 1, \qquad g(1) = 0, \qquad g^{'}(1) = 0 \tag{21}$$

The first two constraints are the limits for the unbroken and fully broken state while the last constraint ensures that $\partial\psi/\partial\phi$ converges to a final value

for the fully broken state $\phi = 1$. Consequently, the parabolic degradation function $g(\phi)$ is introduced as

$$g(\phi) = (1 - \phi)^2 + k \tag{22}$$

where k is a parameter chosen to be as small as possible such that the system of equations are kept well-conditioned for the partly-broken state.

1.2.2. Fracture surface energy

In the total potential energy functional (13), the fracture energy dissipated by the formation of a crack can be written as

$$\Psi^s(\phi) = \int_\Omega G_c \gamma(\phi, \nabla\phi)\, \mathrm{d}V \tag{23}$$

where the parameter G_c is the critical Griffith-type energy release rate of the solid material per unit area. By using the formulation of the stored and fracture energy outlined above, the total potential energy functional can be written as

$$\Psi(\phi, \mathbf{u}) = \int_\Omega \left\{ \left[(1 - \phi)^2 + k\right] \psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{2\ell}\phi^2 + \frac{\ell}{2}|\nabla\phi|^2 \right] \right\} \mathrm{d}V \tag{24}$$

1.3. Governing balance equations of coupled problem

1.3.1. Basic fields and boundary conditions

With the constitutive formulation of the total potential energy in a fracturing solid outlined above, one can now formulate the governing equations. These equations determine the *fracture phase field* ϕ and the *displacement field* $\mathbf{u}$ of the solid introduced in (14) and (15), respectively. With respect to the displacement field, the outer surface of the body is decomposed into a part $\partial\Omega_u$, where the displacement is prescribed by Dirichlet-type boundary conditions

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\mathrm{D}(\mathbf{x}, t) \quad \textit{at} \quad \mathbf{x} \in \partial\Omega_u \tag{25}$$

and into a part $\partial\Omega_h$, where the traction $\mathbf{h}$ is prescribed by Neumann-type boundary conditions (see Fig. 3a). With respect to the fracture phase field, a cracked region can be prescribed through the Dirichlet-type boundary condition

$$\phi(\mathbf{x}, t) = 1 \quad \textit{at} \quad \mathbf{x} \in \Gamma_\mathrm{D} \tag{26}$$

where Γ_{D} is a possible given sharp crack surface inside the solid Ω (see Fig. 3b). The crack phase field ϕ is considered to be driven by the displacement field $\mathbf{u}$ of the solid. Consequently, no prescribed external loading is considered corresponding to the crack phase field ϕ. The external mechanical loading is then defined by the variation of the external work increment as

$$\delta W_{ext} = \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} \, \mathrm{d}V + \int_{\partial \Omega_h} \mathbf{h} \cdot \delta \mathbf{u} \, \mathrm{d}A \tag{27}$$

where $\mathbf{b}$ is a prescribed body force field per unit volume while $\mathbf{h}$ is a boundary traction field per unit area. Furthermore, the variation of the internal potential energy increment is given by

$$\partial W_{int} = \partial \Psi(\phi, \mathbf{u}) = \left(\frac{\partial \Psi}{\partial \boldsymbol{\epsilon}} \right) : \delta \boldsymbol{\epsilon} + \left(\frac{\partial \Psi}{\partial \phi} \right) \delta \phi \tag{28}$$

which for the case of (24) yields

$$\partial W_{int} = \int_{\Omega} \left\{ \boldsymbol{\sigma} \delta \boldsymbol{\epsilon} - 2(1-\phi) \delta \phi \psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{\ell} \phi \delta \phi + \ell \nabla \phi \cdot \nabla \delta \phi \right] \right\} \mathrm{d}V \tag{29}$$

where the Cauchy stresses,

$$\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\epsilon}} = g(\phi) \boldsymbol{\sigma}_{\mathbf{0}} = \left[(1-\phi)^2 + k \right] \boldsymbol{\sigma}_{\mathbf{0}} \tag{30}$$

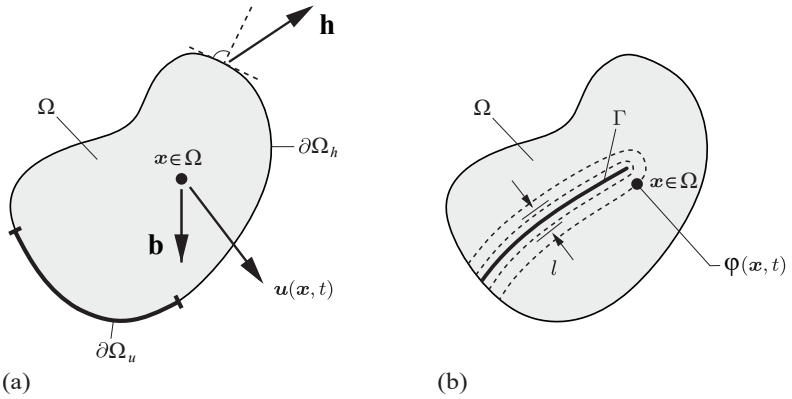

Figure 3: Two-field approach of phase field-type crack propagation in deformable solids. Adapted from [5]. The displacement field $\mathbf{u}$ is constrained by the Dirichlet- and Neumann-type boundary conditions $\mathbf{u} = \mathbf{u}_{\mathrm{D}}$ on $\partial \Omega_u$ and $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h}$ on $\partial \Omega_h$. (b) The crack phase field ϕ is constrained by the Dirichlet- and Neumann-type boundary conditions $\phi = 1$ on Γ and $\nabla \phi \cdot \mathbf{n} = 0$ on $\partial \Omega$.

are given in terms of the stress tensor of the undamaged solid

$$\boldsymbol{\sigma_0} = \mathbf{C_0}\,\boldsymbol{\epsilon} \tag{31}$$

1.3.2. Coupled balances

In order to derive the weak and strong form of the governing equations, the equilibrium of the external and internal virtual works is imposed for the quasi-static process

$$\partial W_{int} - \partial W_{ext} = 0 \tag{32}$$

which by inserting (27) and (29) yields

$$\begin{aligned} &\int_\Omega \left\{ \boldsymbol{\sigma}\delta\boldsymbol{\epsilon} - 2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + G_c\left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi\cdot\nabla\delta\phi\right]\right\} \mathrm{d}V \\ &\quad - \int_\Omega \mathbf{b}\cdot\delta\mathbf{u}\,\mathrm{d}V + \int_{\partial\Omega_h} \mathbf{h}\cdot\delta\mathbf{u}\,\mathrm{d}A = 0 \end{aligned} \tag{33}$$

for all admissible $\delta\phi$ and $\delta\mathbf{u}$ of the phase field and the displacement field, which satisfy the homogeneous form of the Dirichlet-type boundary conditions. Application of the Gauss theorem gives

$$\begin{aligned} &\int_\Omega \left\{ -\left[\mathrm{Div}\left[\boldsymbol{\sigma}\right] + \mathbf{b}\right]\cdot\delta\mathbf{u} - \left[2(1-\phi)\psi_0(\boldsymbol{\epsilon}) - G_c\left[\frac{1}{\ell}\phi - \mathrm{Div}\left[\ell\nabla\phi\right]\right]\right]\delta\phi\right\} \mathrm{d}V \\ &\quad + \int_{\partial\Omega_h} \left[\boldsymbol{\sigma}\cdot\mathbf{n} - \mathbf{h}\right]\cdot\delta\mathbf{u}\,\mathrm{d}A + \int_{\partial\Omega} \left[G_c\ell\nabla\phi\cdot\mathbf{n}\right]\delta\phi\,\mathrm{d}A = 0 \end{aligned} \tag{34}$$

where $\mathbf{n}$ denotes the outward unit vector normal to the surface $\partial\Omega$. Thus, this leads to the strong form of the governing balance equations of the coupled problem

$$\begin{aligned} \mathrm{Div}\left[\boldsymbol{\sigma}\right] + \mathbf{b} &= \mathbf{0} \\ G_c\left[\frac{1}{\ell}\phi - \ell\Delta\phi\right] - 2(1-\phi)\psi_0(\boldsymbol{\epsilon}) &= 0 \end{aligned} \tag{35}$$

along with the Neumann-type boundary conditions

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} \quad \text{on} \quad \partial\Omega_h \qquad \text{and} \qquad \nabla\phi \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega \tag{36}$$

$(35)_1$ is the macroscopic equilibrium condition while $(35)_2$ determines the evolution of the phase field, where $\Delta\phi$ refers to the Laplacian of the phase field.

2. Finite element implementation

2.1. Finite element discretization of variational principles

In order to obtain numerical solutions of the coupled system of partial differential equations (35) using the finite element method, it is more convenient to work with the weak form

$$\begin{gathered} \int_\Omega \{\boldsymbol{\sigma}\delta\boldsymbol{\epsilon} - \mathbf{b}\cdot\delta\mathbf{u}\}\,\mathrm{d}V + \int_{\partial\Omega_h} \mathbf{h}\cdot\delta\mathbf{u}\,\mathrm{d}A = 0 \\ \int_\Omega \left\{-2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + G_c\left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi\cdot\nabla\delta\phi\right]\right\}\mathrm{d}V = 0 \end{gathered} \tag{37}$$

Using Voigt-notation in a 2D space, the displacement field $\mathbf{u}$ and the phase field ϕ can be discretized as

$$\mathbf{u} = \sum_{i=1}^{m} \mathbf{N}_i^{\mathbf{u}}\mathbf{u}_i \qquad \text{and} \qquad \phi = \sum_{i=1}^{m} N_i\phi_i \tag{38}$$

where the shape function matrix is expressed as

$$\mathbf{N}_i^{\mathbf{u}} = \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \tag{39}$$

Here, N_i denotes the shape function associated with node i, m is the total number of nodes per element, and $\mathbf{u}_i = \{u_x, u_y\}^T$ and ϕ_i are the displacement and phase field values at node i. Consequently, the corresponding derivatives can be discretized as

$$\boldsymbol{\epsilon} = \sum_{i=1}^{m} \mathbf{B}_i^{\mathbf{u}}\mathbf{u}_i \qquad \text{and} \qquad \nabla\phi = \sum_{i=1}^{m} \mathbf{B}_i^{\phi}\phi_i \tag{40}$$

where $\boldsymbol{\epsilon} = \{\epsilon_{xx}, \epsilon_{yy}, \epsilon_{xy}\}^T$. The strain-displacement matrices are expressed as

$$\mathbf{B}_i^{\mathbf{u}} = \begin{bmatrix} N_{i,x} & 0 \\ 0 & N_{i,y} \\ N_{i,y} & N_{i,x} \end{bmatrix} \qquad \text{and} \qquad \mathbf{B}_i^{\phi} = \begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} \tag{41}$$

where $N_{i,x}$ and $N_{i,x}$ are the derivatives of the corresponding shape function with respect to x and y, respectively. Similarly, the virtual quantities $\delta\mathbf{u}$ and $\delta\phi$ and their derivatives can be discretized as

$$\begin{aligned} \delta\mathbf{u} &= \sum_{i=1}^{m} \mathbf{N}_i^{\mathbf{u}} \delta\mathbf{u}_i \qquad \text{and} \qquad \delta\phi = \sum_{i=1}^{m} N_i \delta\phi_i \\ \delta\boldsymbol{\epsilon} &= \sum_{i=1}^{m} \mathbf{B}_i^{\delta\mathbf{u}} \mathbf{u}_i \qquad \text{and} \qquad \nabla\delta\phi = \sum_{i=1}^{m} \mathbf{B}_i^{\phi} \delta\phi_i \end{aligned} \tag{42}$$

Using the above expressions and due to the fact that (37) must hold for arbitrary values of $\delta\mathbf{u}$ and $\delta\phi$, the discrete equation corresponding to the equilibrium condition can be expressed as the following residual with respect to the displacement field

$$\mathbf{r}_i^{\mathbf{u}} = \int_{\Omega} \left[(1-\phi)^2 + k\right] (\mathbf{B}_i^{\mathbf{u}})^T \boldsymbol{\sigma}_{\mathbf{0}} \, \mathrm{d}V - \int_{\Omega} (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{b} \, \mathrm{d}V - \int_{\partial\Omega_h} (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{h} \, \mathrm{d}A \tag{43}$$

Similarly, the residual with respect to the evolution of the crack phase field can be expressed as

$$r_i^{\phi} = \int_{\Omega} \left\{ -2(1-\phi) N_i \, \psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{\ell} N_i \, \phi + \ell (\mathbf{B}_i^{\phi})^T \nabla\phi \right] \right\} \, \mathrm{d}V \tag{44}$$

In order to obtain the solutions for which $\mathbf{r}^{\mathbf{u}} = \mathbf{0}$ and $\mathbf{r}^{\phi} = \mathbf{0}$ and due to the fact that the corresponding residuals are nonlinear, an incremental-iterative scheme using Newton-Raphson method is employed

$$\begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t+\Delta t} = \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t} - \begin{bmatrix} \mathbf{K}^{\mathbf{uu}} & \mathbf{K}^{\mathbf{u}\phi} \\ \mathbf{K}^{\phi\mathbf{u}} & \mathbf{K}^{\phi\phi} \end{bmatrix}_{t}^{-1} \begin{Bmatrix} \mathbf{r}^{\mathbf{u}} \\ \mathbf{r}^{\phi} \end{Bmatrix}_{t} \tag{45}$$

in which the tangent stiffness matrices are calculated as

$$\begin{aligned}
\mathbf{K}_{ij}^{\mathbf{uu}} &= \frac{\partial \mathbf{r}_i^{\mathbf{u}}}{\partial \mathbf{u}_j} = \int_\Omega \left[(1-\phi)^2 + k\right] (\mathbf{B}_i^{\mathbf{u}})^T \mathbf{C_0}\, \mathbf{B}_j^{\mathbf{u}}\, \mathrm{d}V \\
\mathbf{K}_{ij}^{\mathbf{u}\phi} &= \frac{\partial \mathbf{r}_i^{\mathbf{u}}}{\partial \phi_j} = \int_\Omega -2\,(1-\phi)\, (\mathbf{B}_i^{\mathbf{u}})^T \boldsymbol{\sigma_0}\, N_j\, \mathrm{d}V \\
\mathbf{K}_{ij}^{\phi\mathbf{u}} &= \frac{\partial r_i^{\phi}}{\partial \mathbf{u}_j} = \int_\Omega -2\,(1-\phi)\, N_i\, \boldsymbol{\sigma_0}^T\, \mathbf{B}_j^{\mathbf{u}}\, \mathrm{d}V \\
\mathbf{K}_{ij}^{\phi\phi} &= \frac{\partial r_i^{\phi}}{\partial \phi_j} = \int_\Omega \left\{ \left[2\psi_0(\boldsymbol{\epsilon}) + \frac{G_c}{\ell} \right] N_i N_j + G_c \ell (\mathbf{B}_i^{\phi})^T (\mathbf{B}_j^{\phi}) \right\} \mathrm{d}V
\end{aligned} \tag{46}$$

The displacement field $\mathbf{u}$ and the phase field ϕ in the coupled system (35) can now be solved simultaneously as fully coupled *monolithic* fields. It should be noted that above system of equations does not guarantee the irreversibility of the evolution of the crack phase field, i.e.

$$\phi_{t+\Delta t} \geq \phi_t \tag{47}$$

However, such a constraint can approximately be enforced by implementing a penalty term in the crack phase field equations as described in [5]. This comes at the cost of adding (artificial) parameters to the model.

2.2. Staggered approach

The *monolithic* model tends to encounter convergence problems in the cases where unstable crack propagation is present. When the crack starts to grow, the stress field rearranges due to the newly degraded stiffness matrix. Consequently, the implicit solver fails to find a stable equilibrium solution. In order to obtain a stable implicit model, the displacement field u and the phase field ϕ in the coupled system (35) can be solved separately as sequentially coupled *staggered* fields. It represents an operator split algorithm in which the phase field and the displacement field are updated sequentially. First, consider the minimization problem of the internal potential energy with respect to the phase field. It yields the strong form of the evolution of the crack phase field along with the Neumann-type boundary condition $(36)_2$

$$G_c \left[\frac{1}{\ell}\phi - \ell \Delta \phi \right] - 2(1-\phi)H = 0 \tag{48}$$

in which H is introduced as so-called history variable field

$$H = \begin{cases} \psi_0(\boldsymbol{\epsilon}) & \text{if } \psi_0(\boldsymbol{\epsilon}) > H_t \\ H_t & \text{otherwise} \end{cases} \tag{49}$$

where H_t is the previously calculated energy obtained in history in terms of the displacement field $\mathbf{u}_t$ at time increment t. The calculation of the current history field is the main idea for the algorithmic split of the coupled equations. Being in fact a variable field which couples the phase field and the displacement field in the staggered scheme. Furthermore, it enforces the irreversibility of the evolution of the crack phase-filed, i.e. the growth constraint (47). Consequently, the history field satisfies the Kuhn-Tucker conditions [4],

$$\psi_0 - H \leq 0, \qquad \dot{H} \geq 0, \qquad \dot{H}(\psi_0 - H) = 0 \tag{50}$$

for both loading and unloading case. Thus, no penalty term is needed in contrast to the monolithic approach. In the same manner as outlined in section 2.1, the residual corresponding to the evolution of the crack phase field is then as follows

$$r_i^{\phi} = \int_{\Omega} \left\{ -2(1-\phi) N_i \, H + G_c \left[\frac{1}{\ell} N_i \, \phi + \ell (\mathbf{B}_i^{\phi})^T \nabla \phi \right] \right\} \mathrm{d}V \tag{51}$$

Similarly, the residual corresponding to the displacement field is then as follows

$$\mathbf{r}_i^{\mathbf{u}} = \int_{\Omega} \left[(1-\phi)^2 + k \right] (\mathbf{B}_i^{\mathbf{u}})^T \boldsymbol{\sigma}_{\mathbf{0}} \, \mathrm{d}V - \int_{\Omega} (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{b} \, \mathrm{d}V - \int_{\partial \Omega_h} (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{h} \, \mathrm{d}A \tag{52}$$

Employing the Newton-Raphson method, the following system of equations can be solved iteratively

$$\begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t+\Delta t} = \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t} - \begin{bmatrix} \mathbf{K}^{\mathbf{uu}} & 0 \\ 0 & \mathbf{K}^{\phi\phi} \end{bmatrix}_{t}^{-1} \begin{Bmatrix} \mathbf{r}^{\mathbf{u}} \\ \mathbf{r}^{\phi} \end{Bmatrix}_{t} \tag{53}$$

in which the tangent stiffness matrices are calculated as

$$\begin{aligned} \mathbf{K}_{ij}^{\mathbf{uu}} &= \frac{\partial \mathbf{r}_i^{\mathbf{u}}}{\partial \mathbf{u}_j} = \int_{\Omega} \left[(1-\phi)^2 + k \right] (\mathbf{B}_i^{\mathbf{u}})^T \mathbf{C}_{\mathbf{0}} \, \mathbf{B}_j^{\mathbf{u}} \, \mathrm{d}V \\ \mathbf{K}_{ij}^{\phi\phi} &= \frac{\partial r_i^{\phi}}{\partial \phi_j} = \int_{\Omega} \left\{ \left[2H + \frac{G_c}{\ell} \right] N_i N_j + G_c \ell (\mathbf{B}_i^{\phi})^T (\mathbf{B}_j^{\phi}) \right\} \mathrm{d}V \end{aligned} \tag{54}$$

The new formulation results in above-decoupled system of equations for the sequential update of the phase field and the displacement field. Then, the staggered scheme for Phase-field model is outlined as follows

1. The displacement, phase and history fields $\mathbf{u}_t$, ϕ_t and H_t are known at the previous time increment t. Then, the prescribed load vectors $\mathbf{b}$ and $\mathbf{h}$ are updated at the current time increment $t + \Delta t$

2. The current phase field $\phi_{t+\Delta t}$ and the current displacement field $\mathbf{u}_{t+\Delta t}$ are now computed using the corresponding linear algebraic Euler equations

$$\mathbf{K}_t^{\phi\phi} \boldsymbol{\phi}_{t+\Delta t} = -\mathbf{r}_t^{\phi} \tag{55}$$

$$\mathbf{K}_t^{\mathbf{uu}} \mathbf{u}_{t+\Delta t} = -\mathbf{r}_t^{\mathbf{u}} \tag{56}$$

where $\mathbf{K}_t^{\phi\phi}$ and $\mathbf{r}_t^{\phi}$ are built from H_t, and $\mathbf{K}_t^{\mathbf{uu}}$ and $\mathbf{r}_t^{\mathbf{u}}$ are built from ϕ_t.

The use of a Newton-Raphson backward Euler scheme implies that the numerical solution is unconditionally stable, i.e., achieving convergence implies attaining the equilibrium solution. Hence, the solution is independent of the time increment, although small load increments may be needed to achieve convergence if there is a sudden change in the solution (as in, e.g., unstable crack propagation). In the *staggered* scheme adopted here, the phase and the displacement fields are solved - within a Backward Euler approach - independently. This is unlike the *monolithic* case, where convergence is fulfilled for both coupled fields at the same time, retaining the unconditional stability. The price to pay in the *monolithic* approach is the need to include a penalty term (introducing additional parameters into the model) and difficulties to achieve convergence in the very computationally demanding problem of unstable crack propagation. By solving for the phase and the displacement fields independently, the *staggered* scheme can track the equilibrium solution in a semi-implicit manner, leading to a more robust numerical implementation. One should, however, perform a time increment sensitivity study.

The specific staggered scheme outlined has proven to be the more robust one and is therefore chosen for our implementation. Files are provided with other two options: (1) a semi-implicit approach where $\mathbf{K}_t^{\mathbf{uu}}$ and $\mathbf{r}_t^{\mathbf{u}}$ are built from $\mathbf{u}_t + \Delta t$ (denoted file B), and (2) an implicit, weakly coupled, approach where $\mathbf{u}_t + \Delta t$ and $H_{t+\Delta t}$ are used to build the linearized system through

an extrapolation procedure (denoted file A) - differences and similarities are discussed at large in Appendix A. Msekh et al. [15] implemented the phase field fracture method in Abaqus by means of an implicit approach. A more robust approach is the one presented by Molnar and Gravouil [16]; however, they make use of a cumbersome multi-layer system that hinders the implementation.

2.3. Numerical implementation in Abaqus

The phase field model is implemented by means of Abaqus UEL subroutine which allows for user-defined computation of the element tangent stiffness matrices and the nodal force vectors. We consider isoparametric 2D quadrilateral elements (linear and quadratic) with 3 degrees of freedom per node, i.e. u_x, u_y and ϕ, and four integration points. The extension to a three dimensional case is straightforward.

A number of quantities are stored as solution-dependent state variables `SVARS` to use semi-implicit staggered schemes and ease the extension to history-dependent problems (e.g., plasticity). These are shown in Table 1. The stress variables refer to the undamaged stress tensor $\boldsymbol{\sigma}_0$.

Variable	SVARS numbering
Axial stresses - σ_{11} , σ_{22} , σ_{33}	SVARS(1), SVARS(2), SVARS(3)
Shear stress - σ_{12}	SVARS(4)
Axial strains - ϵ_{11} , ϵ_{22} , ϵ_{33}	SVARS(5), SVARS(6), SVARS(7)
Shear strain - ϵ_{12}	SVARS(8)
History variable field - H	SVARS(9)
Crack phase field - ϕ	SVARS(10)

Table 1: List of solution dependent state variables for the UEL.

The use of user element subroutines has the drawback that integration point variables cannot be visualized in Abaqus/Viewer. This limitation is intrinsic to the fact that the only information that Abaqus requests from the UEL subroutine are the stiffness matrix and the right-hand side nodal force vector - the magnitude of the stresses and the strains, as well as the choice of shape functions, is information that is not available as output. To overcome this limitation we here make use of an auxiliary dummy mesh consisting of

standard Abaqus elements that resemble the user defined element in terms of number of nodes and integration points (i.e., CPE4 or CPE8R). The material response at each integration point in the auxiliary mesh is defined using a user material subroutine (UMAT), which enables the user to define the constitutive matrix and the stresses from the strain values. In this auxiliary mesh, the stress components and the constitutive matrix are made equal to zero (i.e., they have no influence in the solution of the global system). The data from our UEL that we want to observe in Abaqus/Viewer is stored in a Fortran module, which allows transferring to the UMAT subroutine. In the UMAT the information is passed to the built-in array `STATEV` for each corresponding element and integration point. If SDV variables are requested as Field Output we would be able to visualize the results. Table 2 shows the equivalence between model variables and SDVs.

Variable	SDVs numbering
Axial stresses - σ_{11} , σ_{22} , σ_{33}	SDV1, SDV2, SDV3
Shear stress - σ_{12}	SDV4
Axial strains - ϵ_{11} , ϵ_{22} , ϵ_{33}	SDV5, SDV6, SDV7
Shear strain - ϵ_{12}	SDV8
Crack phase field - ϕ	SDV9

Table 2: List of solution dependent state variables.

2.4. Usage instructions

The first step is to create the model in Abaqus/CAE. The procedure is the same as with standard Abaqus models with the following subtleties:

- The material has to be defined as a user material with 9 solution-dependent variables. (General → Depvar: 9 & General → User Material - Mechanical Constants: 0).
- SDV, Solution dependent state variables, have to be requested as Field Output (as well as displacement, reaction forces and other relevant quantities). (Field Output Request - State/Field/User/Time: SDV, Solution dependent state variables)
- In the (Static, General) Step definition we will define the incrementation type as "Fixed" (as opposed to "Automatic") to use a constant

time increment. This is due to the semi-implicit staggered scheme adopted (see discussion in Section 2.2).

- The mesh has to be very refined in the expected crack propagation area. As discussed in our publication [12], the characteristic element size has to be at least 5 times smaller than ℓ to resolve the fracture process zone. If the crack path is unknown a common strategy is to start with a coarser uniform mesh and refine in subsequent calculations. Use as element type CPE4 (or CPE8R if you run the analysis with the quadratic element file).

Once the model has been developed, we create a job and write the input file (Right click on the Job name and click "Write Input"). A few modifications have to be done to the input file to define the user element, the use of a code editor like Notepad++ is recommended. First, we create the dummy visualization mesh. For this purpose we use the Matlab script VirtualMesh.m, which is part of the Abaqus2Matlab package [17]. Running VirtualMesh.m on the same folder as the input file (Job-1.inp) will create a new file (VisualMesh.inp) with the element connectivity of the visualization mesh.

The first step is to replace the element type,

```
*Element, type=CPE4
```

with the user element definition,

```
*User element, nodes=4, type=U1, properties=5, coordinates=2, var=40
1,2
1,3
*ELEMENT, TYPE=U1, ELSET=SOLID
```

where we have defined the number of nodes (linear version), the number of properties that will be defined in the input file, the number of coordinates (2D), and the number of `SVARS` (10 per integration point). We have defined the ordering of the DOFs in a way that (53) corresponds to the element system (as opposed to the node system). Thus, the variable `U` contains the components: u_x^1, u_y^1, u_x^2, u_y^2, u_x^3, u_y^3, u_x^4, u_y^4, ϕ^1, ϕ^2, ϕ^3 and ϕ^4. Accordingly, if one wishes to prescribe a cracked region through the phase field parameter, the boundary condition $\phi = 1$ should be enforced on the DOF 3.

After the element connectivity list one inserts,

```
*UEL PROPERTY, ELSET=SOLID
210000., 0.3, 0.025, 2.7, 1e-07
*Element, type=CPE4, elset=Visualization
```

and immediately afterwards the visualization connectivity list (i.e., the content of the file VisualMesh.inp created by the Matlab script). Here, we have defined the user element properties following Table 3. Throughout our model we employ SI (mm) units.

UEL PROPERTY	Description
PROPS(1)	E - Young's modulus [MPa]
PROPS(2)	ν - Poisson's ratio
PROPS(3)	ℓ - Phase field length parameter [mm]
PROPS(4)	G_c - Critical energy release rate [MPa mm]
PROPS(5)	k - well-conditioning parameter

Table 3: List of user element properties.

And finally, note that, since we have defined our dummy connectivity list within the element set "Visualization", we need to modify the Section definition,

```
*Solid Section, elset=Set-1, material=Material-1
```

to change the name of the element set,

```
*Solid Section, elset=Visualization, material=Material-1
```

Additionally, one should note that a Fortran module has been defined in the first lines of the subroutine for visualization purposes. One has to be sure that the first dimension of the variable `UserVar` is larger than the total number of elements.

2.5. Representative results

We consider as benchmark the case of a square plate with a horizontal crack placed at the middle point of the left side of the plate. The geometric set-up as well as the boundary conditions are illustrated in Fig. 4. The bottom side is fixed while the top edge is moved vertically. Young's modulus is chosen to be $E = 210000$ MPa, Poisson's ratio $\nu = 0.3$ and critical energy

release rate $G_c = 2.7$ MPa mm. The load is applied by prescribing a constant total displacement of $u = 0.01$ mm.

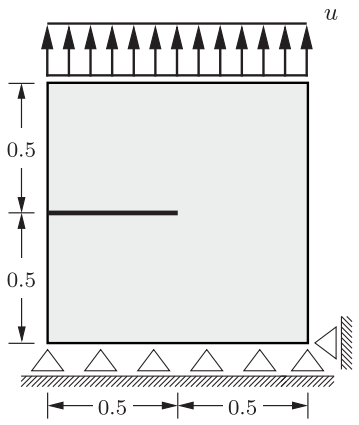

Figure 4: Notched square plate subjected to tension test, geometry and boundary conditions.

Since the goal is to provide an example file, a coarse mesh is adopted to allow for a rapid simulation (the job finishes in minutes). A total of 5409 quadrilateral elements are employed, with the characteristic element length along the crack propagation path being equal to $h = 0.005$ mm. We adopt a length scale that is 8 times larger than h to ensure mesh-independent results, $\ell = 0.04$ mm. To run the calculation type in the command line:

```
abaqus job=Job-1 user=PhaseField.for
```

Linux users may have to change the extension of the subroutine, converting the `PhaseField.for` file to `PhaseField.f`. The Python script `ResultsQ4.py` can be used to obtain relevant results. Representative results are shown in Figs. 5 and 6. Specifically, Fig. 5 shows the force versus displacement curve obtained. As observed in the figure, damage brings in an important drop in the load, with the crack propagating in an unstable manner across the specimen. Computations persist beyond the point of almost complete breakage. The resulting fracture patterns at different steps are illustrated in Fig. 6. Blue and red colors correspond to the completely intact and the fully broken state of the material, respectively. The response is not symmetric, as the lower bound is fully clamped, and the crack is rather diffuse, as expected given the choice of ℓ. A quantitative agreement with the results of Miehe

et al. [5] can be adopted by choosing the same value of ℓ and refining the mesh (see details in our paper [12]). A time sensitivity analysis should also be conducted (a relatively large time increment is chosen to enable a fast demonstration).

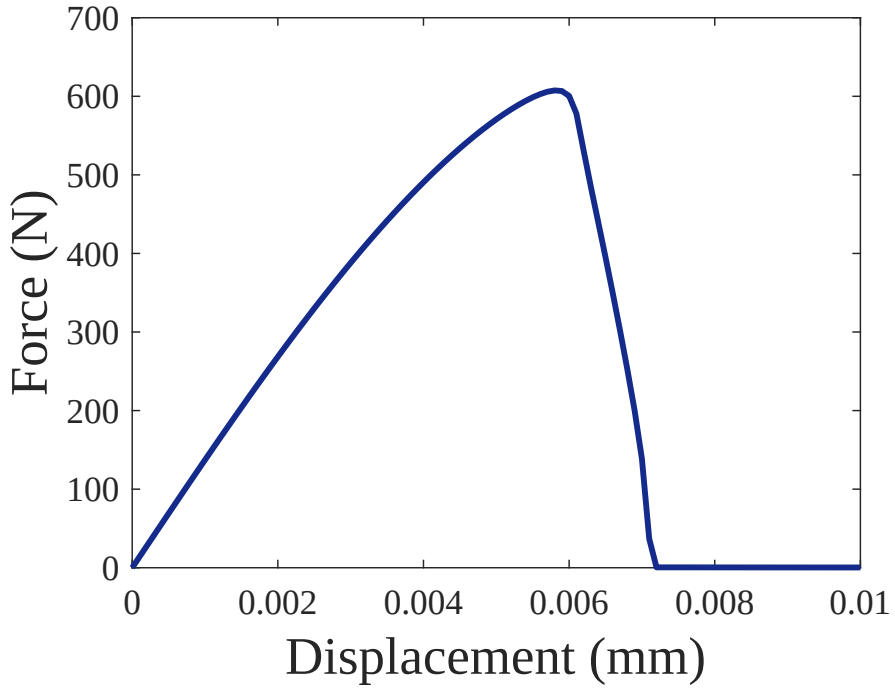

Figure 5: Load-deflection curve obtained in the simple benchmark of a cracked square plate subjected to tension.

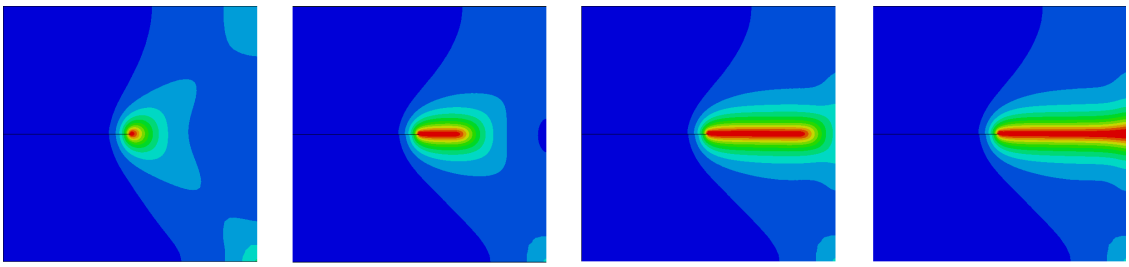

Figure 6: Notched square plate subjected to tension test. Fracture pattern at a displacement of (a) $u = 0.06$ mm, (b) $u = 0.065$ mm, (c) $u = 0.07$ mm and (d) $u = 0.075$ mm.

In addition to the main subroutine and the input file for this simple boundary value problem, subroutines and input files are provided for other integration schemes and type of elements (see Appendix B).

3. Conclusions

We have provided a robust implementation of the phase field fracture method for the commercial finite element package Abaqus. As discrete methods (see, e.g., [18]), the phase field fracture method requires a refined mesh along the potential crack propagation path to resolve the fracture process zone. However, by decoupling the damage and displacement variables the method is very robust and it is able to model unstable crack propagation without the need of control algorithms [19, 20]. Also, as shown in our journal publication [12], the phase field method is well-suited to deal with arbitrary crack propagation paths. In summary, the method holds promise and we hope that the present implementation will facilitate research in this field.

4. Acknowledgements

E. Martínez-Pañeda acknowledges financial support from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n° 609405 (COFUNDPostdocDTU).

Appendix A. Assessment of staggered schemes in Abaqus

Let us consider the simple case of a displacement-based finite element scheme without any additional fields. We aim at computing the displacement field $\mathbf{u}$ of the following linearised problem,

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{A.1}$$

If our problem is non-linear, we need to prescribe the external load $\mathbf{f}$ incrementally to facilitate convergence of the Newton-Raphson algorithm,

$$\mathbf{K}\Delta\mathbf{u} = \Delta\mathbf{f} \tag{A.2}$$

such that we are subsequently computing the incremental displacement solution. One can readily obtain the total solution at the current time increment by,

$$\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta\mathbf{u} \tag{A.3}$$

Abaqus provides to the UEL subroutine the incremental solution $\Delta\mathbf{u}$ (variable `DU`) and the total solution at the current time increment $\mathbf{u}_{t+\Delta t}$ (variable `U`). From $\Delta\mathbf{u}$ one can readily compute the increments of strain, that are added to the stored strain quantities, and this information is passed to a UMAT subroutine where the stiffness matrices and the stresses are computed. If we are dealing with a history dependent problem (e.g., plasticity), our stiffness matrix will be computed from the updated total strains and stresses.

In non-linear problem, several iterations may be needed to achieve convergence. Abaqus adopts by default an extrapolation scheme by which $\Delta\mathbf{u}_{t+\Delta t}$, needed to build $\mathbf{K}$, will be assumed to be equal to the converged solution of the previous increment $\Delta\mathbf{u}_t$. Thus, for the first iteration we build our stiffness matrix using $\Delta\mathbf{u}_0 (= \Delta\mathbf{u}_t)$. If we do not achieve convergence we add the new solution $\Delta\mathbf{u} = \Delta\mathbf{u}_0 + \Delta\mathbf{u}_1$, update $\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta\mathbf{u}$ and compute $\mathbf{K}$. The process is repeated until convergence is achieved: $\Delta\mathbf{u}_k$, and the residual, are sufficiently small.

In the context of our phase field implementation this implies that we have, at the UEL, the proposed (it may not converge) solution for the current load $\mathbf{u}_{t+\Delta t}$ and $\phi_{t+\Delta t}$. Miehe et al. [5] suggest solving the phase field problem with $\mathbf{K}^{\phi\phi}$ and $\mathbf{r}^{\phi}$ computed from H_t, and then solve the displacement

problem with the recently obtained $\phi_{t+\Delta t}$. One can exactly reproduce this scheme in Abaqus using sub-increments, solving for $\Delta\phi$ and $\Delta\mathbf{u}$ in alternate sub-increments. Another approach, taken here, is to solve at the same time for both $\Delta\phi$ and $\Delta\mathbf{u}$. One can choose to build $\mathbf{K}^{\phi\phi}$ and $\mathbf{r}^{\phi}$ from H_t, as done by Miehe et al. [5], or follow a more implicit procedure where we employ $H_{t+\Delta t}$, as computed from the proposed solution for the current increment $\mathbf{u}_{t+\Delta t}$. This latter implicit approach may consider H_t instead of $H_{t+\Delta t}$, as dictated by the Kuhn-Tucker conditions. We found that convergence improves if H_t is updated during the non-converged increments as well. While this sceme is unconditionally stable, we struggle to achieve convergence in many of our tests; files are nevertheless provided, we refer to it as approach A. Another option, referred to as approach B, is to follow Miehe et al. [5], computing $\mathbf{K}^{\phi\phi}$ and $\mathbf{r}^{\phi}$ from H_t, and build $\mathbf{K}^{\mathbf{uu}}$ and $\mathbf{r}^{\mathbf{u}}$ from the proposed $\phi_{t+\Delta t}$ (1 field, semi-implicit). A third approach (C), more robust, consists in computing $\mathbf{K}^{\phi\phi}$ and $\mathbf{r}^{\phi}$ from H_t, and build $\mathbf{K}^{\mathbf{uu}}$ and $\mathbf{r}^{\mathbf{u}}$ from the converged ϕ_t (2 fields, semi-implicit).

Two additional files are provided, although we never found the need to use them for the computations conducted in [12]. However, they may be useful for studies that entail strong convergence problems. One, denoted as approach D, is the same as the 2 fields semi-implicit approach (C) but we update H_t even if the increment does not converge. This has proven to help in some of our convergence tests. The other one is a fully forward Euler approach with equilibrium correction; this is referred to as approach E. While it seems to work very well, one should be very careful to ensure that the increment is sufficiently small such that we are not going away from the equilibrium solution. In our tests we found that the explicit version works best with approach B - semi-implicit, one field.

Appendix B. List of files

Main folder

Job-1.inp - Input file for the benchmark problem of a cracked square subjected to tension.

PhaseField.for - UEL Subroutine with the phase field fracture model. 4-node element and semi-implicit integration scheme (2 fields, C).

ResultsQ4.py - Python script to automatically extract the load displacement curve and show contours.

ExtraFiles folder

Job-1.inp - Example of input file as obtained from Abaqus/CAE, to be read by VisualMesh.m.

VisualMesh.m - Matlab script from [17] to create the element connectivity list of the visualization mesh.

Job-1a.inp - Input file for the implicit case (A), to be used with PhaseFieldQ4a.for.

Job-1b.inp - Input file for the semi-implicit (1 field) case (B), to be used with PhaseFieldQ4b.for.

Job-1d.inp - Input file for the semi-implicit (2 fields) case storing non-converged H_t (D), to be used with PhaseFieldQ4d.for.

Job-1e.inp - Input file for the forward Euler case with residual correction (E), to be used with PhaseFieldQ4b.for.

PhaseFieldQ4a.for - Subroutine for the implicit case (A).

PhaseFieldQ4b.for - Subroutine for the semi-implicit case (1 field, B).

PhaseFieldQ4d.for - Subroutine for the semi-implicit (2 fields) case storing non-converged H_t (D).

Q8 folder

Job-1c.inp - Input file for quadratic quadrilateral elements (semi-implicit, 2 fields, C).

PhaseFieldQ8c.for - Subroutine for the quadratic element case (semi-implicit, 2 fields, C).

References

[1] G. Francfort, J.-J. Marigo, Revisiting brittle fracture as an energy minimization problem, Journal of the Mechanics and Physics of Solids 46 (8) (1998) 1319–1342.

[2] B. Bourdin, G. A. Francfort, J. J. Marigo, Numerical experiments in revisited brittle fracture, Journal of the Mechanics and Physics of Solids 48 (4) (2000) 797–826.

[3] B. Bourdin, G. A. Francfort, J. J. Marigo, The variational approach to fracture, Springer Netherlands, 2008.

[4] C. Miehe, F. Welshinger, M. Hofacker, Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations, International Journal for Numerical Methods in Engineering 83 (2010) 1273–1311.

[5] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, Computer Methods in Applied Mechanics and Engineering 199 (45-48) (2010) 2765–2778.

[6] M. J. Borden, T. J. R. Hughes, C. M. Landis, A. Anvari, I. J. Lee, A phase-field formulation for fracture in ductile materials: Finite deformation balance law derivation, plastic degradation, and stress triaxiality effects, Computer Methods in Applied Mechanics and Engineering 312 (2016) 130–166.

[7] C. Miehe, F. Aldakheel, A. Raina, Phase field modeling of ductile fracture at finite strains: A variational gradient-extended plasticity-damage theory, International Journal of Plasticity 84 (2016) 1–32.

[8] A. Mikelic, M. Wheeler, T. Wick, A phase-field method for propagating fluid-filled fractures coupled to a surrounding porous medium, Multiscale Modeling and Simullation 13 (2015) 367–398.

[9] Z. A. Wilson, C. M. Landis, Phase-field modeling of hydraulic fracture, Journal of the Mechanics and Physics of Solids 96 (2016) 264–290.

[10] J. Reinoso, M. Paggi, C. Linder, Phase field modeling of brittle fracture for enhanced assumed strain shells at large deformations: formulation and finite element implementation, Computational Mechanics 59 (6) (2017) 981–1001.

[11] V. Carollo, J. Reinoso, M. Paggi, A 3D finite strain model for intralayer and interlayer crack simulation coupling the phase field approach and cohesive zone model, Composite Structures 182 (2017) 636–651.

[12] E. Martínez-Pañeda, A. Golahmar, C. F. Niordson, A phase field formulation for hydrogen assisted cracking, Computer Methods in Applied Mechanics and Engineering 342 (2018) 742–761.

[13] E. Martínez-Pañeda, C. F. Niordson, On fracture in finite strain gradient plasticity, International Journal of Plasticity 80 (2016) 154–167.

[14] E. Martínez-Pañeda, S. Natarajan, S. Bordas, Gradient plasticity crack tip characterization by means of the extended finite element method, Computational Mechanics 59 (5) (2017) 831–842.

[15] M. A. Msekh, J. M. Sargado, M. Jamshidian, P. M. Areias, T. Rabczuk, Abaqus implementation of phase-field model for brittle fracture, Computational Materials Science 96 (PB) (2015) 472–484.

[16] G. Molnar, A. Gravouil, 2D and 3D Abaqus implementation of a robust staggered phase-field solution for modeling brittle fracture, Finite Elements in Analysis and Design 130 (March) (2017) 27–38.

[17] G. Papazafeiropoulos, M. Muñiz-Calvente, E. Martínez-Pañeda, Abaqus2Matlab: A suitable tool for finite element post-processing, Advances in Engineering Software 105 (2017) 9–16.

[18] S. del Busto, C. Betegón, E. Martínez-Pañeda, A cohesive zone framework for environmentally assisted fatigue, Engineering Fracture Mechanics 185 (2017) 210–226.

[19] J. Segurado, J. LLorca, A new three-dimensional interface finite element to simulate fracture in composites, International Journal of Solids and Structures 41 (11-12) (2004) 2977–2993.

[20] E. Martínez-Pañeda, S. del Busto, C. Betegón, Non-local plasticity effects on notch fracture mechanics, Theoretical and Applied Fracture Mechanics 92 (2017) 276–287.