Quasi-Newton implementation of the phase field method for fracture and fatigue in ABAQUS

Philip K. Kristensen[b], Emilio Martínez-Pañeda[a,*]

[a]*Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, 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 and fatigue in Abaqus using the quasi-Newton solution scheme. If using this code for research or industrial purposes, please cite:

P.K. Kristensen, E. Martínez-Pañeda. Phase field fracture modelling using quasi-Newton methods and a new adaptive step scheme. Theoretical and Applied Fracture Mechanics 102446 (2020)

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 topology by means of a scalar damage variable and a diffuse crack representation. This variable

[*] Corresponding author.
Email address: e.martinez-paneda@imperial.ac.uk (Emilio Martínez-Pañeda)

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], fatigue [10, 11], and hydrogen assisted cracking [12, 13], 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 to read about the extension to fatigue or to Section 3.3 for the usage of the files provided.

1.1. Representation of crack surface density function

Alan Arnold Griffith's energy-based analysis of cracks in 1920 is considered to be the birth of the field of fracture mechanics [14]. Consider a cracked solid with strain energy density $\psi(\boldsymbol{\varepsilon})$, which is a function of the strain tensor $\boldsymbol{\varepsilon}$. In the absence of external forces, the variation of the total energy Π due to an incremental increase in the crack area $\mathrm{d}A$ is given by

$$\frac{\mathrm{d}\Pi}{\mathrm{d}A} = \frac{\mathrm{d}\psi(\boldsymbol{\varepsilon})}{\mathrm{d}A} + \frac{\mathrm{d}W_c}{\mathrm{d}A} = 0, \tag{1}$$

where W_c is the work required to create new surfaces. The last term is the so-called critical energy release rate $G_c = \mathrm{d}W_c/\mathrm{d}A$, a material property that characterises the fracture resistance. Griffith's energy balance can be formulated in a variational form as:

$$\Pi = \int_\Omega \psi\left(\boldsymbol{\varepsilon}\right) \mathrm{d}V + \int_\Gamma G_c \, \mathrm{d}\Gamma, \tag{2}$$

with Γ being the crack surface and V denoting the volume of the solid, occupying an arbitrary domain Ω. The crack surface is unknown, hindering minimization of (2). However, an auxiliary variable, the phase field ϕ, can be used to track the crack interface, see Fig. 1. The phase field ϕ is a damage-like variable that takes the values of 0 in an intact material point, and of 1 in a fully cracked material point.

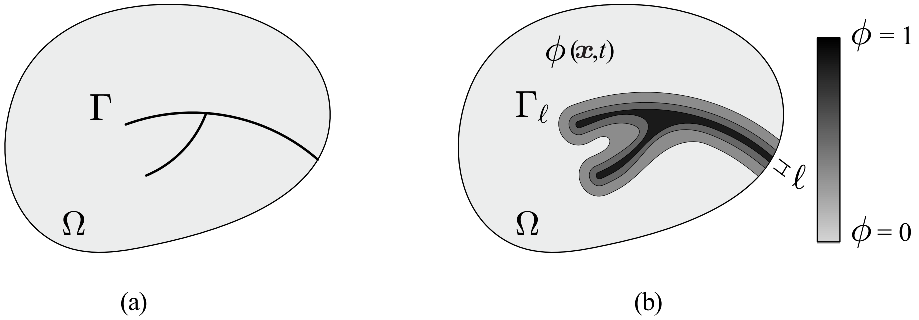

Figure 1: Schematic representation of a solid body with (a) internal discontinuity boundaries, and (b) a phase field approximation of the discrete discontinuities.

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

$$\Pi(\mathbf{u}, \phi) = \Pi^b(\mathbf{u}, \phi) + \Pi^s(\phi) \tag{3}$$

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.

1.1.1. Isotropic degradation of the stored bulk energy

In the total potential energy functional (3), the stored bulk energy can be written as

$$\Pi^b(\mathbf{u}, \phi) = \int_\Omega \psi\left(\boldsymbol{\epsilon}(\mathbf{u}), \phi\right)\, \mathrm{d}V \tag{4}$$

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

$$\psi\left(\boldsymbol{\epsilon}, \phi\right) = g(\phi)\psi_0(\boldsymbol{\epsilon}) \tag{5}$$

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

$$\psi_0(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon}^T \mathbf{C_0} \boldsymbol{\epsilon} \tag{6}$$

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 [15, 16]. Considering the plane strain condition in 2D

$$\mathbf{C}_0 = \tfrac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \tfrac{1-2\nu}{2} \end{bmatrix} \tag{7}$$

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

$$\boldsymbol{\epsilon} = \frac{1}{2}\left[\nabla^T \mathbf{u} + \nabla \mathbf{u}\right] \tag{8}$$

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

$$g(0) = 1, \qquad g(1) = 0, \qquad g'(1) = 0 \tag{9}$$

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

$$g(\phi) = (1-\phi)^2 + k \tag{10}$$

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.

1.1.2. Fracture surface energy

In the total potential energy functional (3), the fracture energy dissipated by the formation of a crack can be written as

$$\Pi^s(\phi) = \int_\Omega G_c \gamma(\phi, \nabla\phi)\, \mathrm{d}V \tag{11}$$

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

$$\Pi(\phi, \mathbf{u})_\ell = \int_\Omega \left\{ \left[(1-\phi)^2 + k \right] \psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{2\ell}\phi^2 + \frac{\ell}{2}|\nabla\phi|^2 \right] \right\} \mathrm{d}V \tag{12}$$

where ℓ is a length scale parameter that governs the size of the fracture process zone and ψ_0 denotes the elastic strain energy of the undamaged solid. The work required to create a cracked surface, Γ, is now expressed as a volume integral, making the problem computationally tractable. As shown by Γ-convergence, the regularized functional Π_ℓ approaches the functional of the discrete crack problem Π for $\ell \to 0$ [17, 18]. The form of the phase field terms related to the surface energy is based on the work by Bourdin *et al.* [2] and the earlier regularization by Ambrosio and Tortorelli of the Mumford-Shah problem in image processing [19]. This surface regularization is commonly referred to as the AT2 model. See Ref. [20] for other choices and a detailed numerical comparison in the context of phase field fracture. Considering the earlier work by Wu *et al.* [21], the superior performance of monolithic quasi-Newton solution strategies is therefore demonstrated for both the PF-CZM and AT2 regularizations; the analysis of the so-called AT1 model [22] remains to be addressed.

1.2. Governing balance equations of coupled problem

1.2.1. 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. 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

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\mathrm{D}(\mathbf{x}, t) \quad at \quad \mathbf{x} \in \partial\Omega_u \tag{13}$$

and into a part $\partial\Omega_h$, where the traction $\mathbf{h}$ is prescribed by Neumann-type boundary conditions (see Fig. 2a). With respect to the fracture phase field, a cracked region can be prescribed through the Dirichlet-type boundary condition

$$\phi(\mathbf{x}, t) = 1 \quad at \quad \mathbf{x} \in \Gamma_\mathrm{D} \tag{14}$$

where Γ_{D} is a possible given sharp crack surface inside the solid Ω (see Fig. 2b). 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

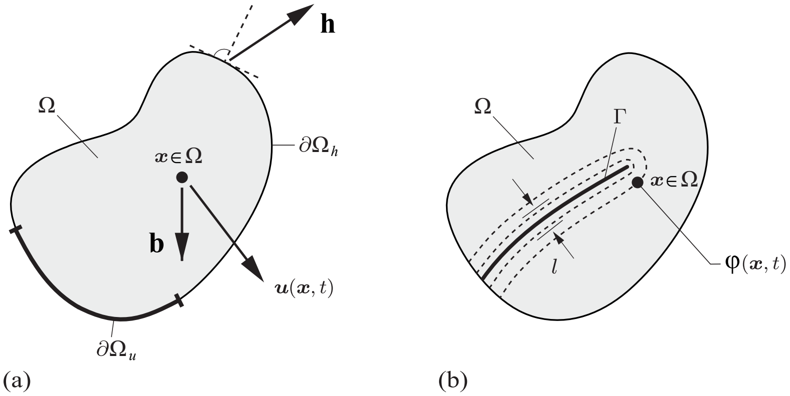

Figure 2: Two-field approach of phase field-type crack propagation in deformable solids. Adapted from [5]. The displacement field $\mathbf{u}$ is constrained by the Dirichlet- and Neumann-type boundary conditions $\mathbf{u} = \mathbf{u}_{\mathrm{D}}$ on $\partial\Omega_u$ and $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h}$ 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 $\partial\Omega$.

$$\delta W_{ext} = \int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} \, \mathrm{d}V + \int_{\partial\Omega_h} \mathbf{h} \cdot \delta\mathbf{u} \, \mathrm{d}A \tag{15}$$

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

$$\partial W_{int} = \partial\Psi(\phi, \mathbf{u}) = \left(\frac{\partial\Psi}{\partial\boldsymbol{\epsilon}}\right) : \delta\boldsymbol{\epsilon} + \left(\frac{\partial\Psi}{\partial\phi}\right) \delta\phi \tag{16}$$

which for the case of (12) yields

$$\partial W_{int} = \int_{\Omega} \left\{ \boldsymbol{\sigma}\delta\boldsymbol{\epsilon} - 2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi \cdot \nabla\delta\phi \right] \right\} \mathrm{d}V \tag{17}$$

where the Cauchy stresses,

$$\boldsymbol{\sigma} = \frac{\partial\Psi}{\partial\boldsymbol{\epsilon}} = g(\phi)\boldsymbol{\sigma_0} = \left[(1-\phi)^2 + k\right] \boldsymbol{\sigma_0} \tag{18}$$

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

$$\boldsymbol{\sigma_0} = \mathbf{C_0}\, \boldsymbol{\epsilon} \tag{19}$$

1.2.2. 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

$$\partial W_{int} - \partial W_{ext} = 0 \tag{20}$$

which by inserting (15) and (17) yields

$$\begin{aligned} &\int_\Omega \left\{ \boldsymbol{\sigma}\delta\boldsymbol{\epsilon} - 2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + G_c \left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi \cdot \nabla\delta\phi\right]\right\} \mathrm{d}V \\ &\quad - \int_\Omega \mathbf{b} \cdot \delta\mathbf{u}\, \mathrm{d}V + \int_{\partial\Omega_h} \mathbf{h} \cdot \delta\mathbf{u}\, \mathrm{d}A = 0 \end{aligned} \tag{21}$$

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

$$\begin{aligned} &\int_\Omega \left\{ - \left[\mathrm{Div}\left[\boldsymbol{\sigma}\right] + \mathbf{b}\right] \cdot \delta\mathbf{u} - \left[2(1-\phi)\psi_0(\boldsymbol{\epsilon}) - G_c\left[\frac{1}{\ell}\phi - \mathrm{Div}\left[\ell\nabla\phi\right]\right]\right]\delta\phi\right\} \mathrm{d}V \\ &\quad + \int_{\partial\Omega_h} \left[\boldsymbol{\sigma}\cdot\mathbf{n} - \mathbf{h}\right] \cdot \delta\mathbf{u}\, \mathrm{d}A + \int_{\partial\Omega} \left[G_c\ell\nabla\phi\cdot\mathbf{n}\right]\delta\phi\, \mathrm{d}A = 0 \end{aligned} \tag{22}$$

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

$$\begin{aligned} \mathrm{Div}\left[\boldsymbol{\sigma}\right] + \mathbf{b} &= \mathbf{0} \\ G_c\left[\frac{1}{\ell}\phi - \ell\Delta\phi\right] - 2(1-\phi)\psi_0(\boldsymbol{\epsilon}) &= 0 \end{aligned} \tag{23}$$

along with the Neumann-type boundary conditions

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} \quad \text{on} \quad \partial\Omega_h \qquad \text{and} \qquad \nabla\phi \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega \tag{24}$$

$(23)_1$ is the macroscopic equilibrium condition while $(23)_2$ determines the evolution of the phase field, where $\Delta\phi$ refers to the Laplacian of the phase field.

2. Extension to fatigue

The framework has been extended to also handle fatigue based on the implementation proposed in [10]. The general idea is centered around the introduction of a cumulative history variable $\bar{\alpha}$, which governs the accumulation of fatigue and propagates the crack by means of a fatigue degradation function $f(\bar{\alpha})$, which locally lowers the fracture energy in the vicinity of the crack. The fatigue degradation function enters the governing equation for the phase field as:

$$\int_\Omega \left\{ -2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + f(\bar{\alpha})G_c \left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi \cdot \nabla\delta\phi \right] \right\} \mathrm{d}V = 0 \tag{25}$$

The cumulative history variable $\bar{\alpha}$ is defined such that it only grows during loading. The simplest formulation proposed in [10] takes the form:

$$\overline{\alpha}(t) = \int_0^t \theta(\alpha\dot{\alpha})|\dot{\alpha}| \, \mathrm{d}\tau, \tag{26}$$

where τ is the pseudo-time and $\theta(\alpha\dot{\alpha})$ is the Heaviside function. The fatigue history variable α is a scalar variable which represents the loading conditions in the material. It is here simply $\alpha = g(\phi)\psi_0$. Finally, a formulation for the fatigue degradation function $f(\bar{\alpha})$ could be the asymptotically vanishing formulation from [10]:

$$f(\overline{\alpha}(t)) = \begin{cases} 1 & \text{if } \overline{\alpha}(t) \leq \alpha_T \\ \left(\dfrac{2\alpha_T}{\overline{\alpha}(t) + \alpha_T} \right) & \text{if } \overline{\alpha}(t) \geq \alpha_T \end{cases}. \tag{27}$$

Here, α_T represents a threshold value, below which the fracture energy remains unaffected. As a first approximation, it is chosen as:

$$\alpha_T = \frac{G_C}{12\ell} \tag{28}$$

All of the chosen formulations above are first guesses to capture the fatigue phenomenon. Other choices could most likely be made to better represent the actual physics and to include features such as a fatigue threshold.

3. Finite element implementation

3.1. Finite element discretization of variational principles

In order to obtain numerical solutions of the coupled system of partial differential equations (23) using the finite element method, it is more convenient to work with the weak form:

$$\begin{aligned} \int_{\Omega} \{\boldsymbol{\sigma}\delta\boldsymbol{\epsilon} - \mathbf{b}\cdot\delta\mathbf{u}\}\,\mathrm{d}V + \int_{\partial\Omega_h} \mathbf{h}\cdot\delta\mathbf{u}\,\mathrm{d}A = 0 \\ \int_{\Omega} \left\{-2(1-\phi)\delta\phi\psi_0(\boldsymbol{\epsilon}) + f(\overline{\alpha})G_c\left[\frac{1}{\ell}\phi\delta\phi + \ell\nabla\phi\cdot\nabla\delta\phi\right]\right\}\mathrm{d}V = 0 \end{aligned} \tag{29}$$

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

$$\mathbf{u} = \sum_{i=1}^{m} \mathbf{N}_i^{\mathbf{u}}\mathbf{u}_i \qquad \text{and} \qquad \phi = \sum_{i=1}^{m} N_i\phi_i \tag{30}$$

where the shape function matrix is expressed as

$$\mathbf{N}_i^{\mathbf{u}} = \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \tag{31}$$

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

$$\boldsymbol{\epsilon} = \sum_{i=1}^{m} \mathbf{B}_i^{\mathbf{u}}\mathbf{u}_i \qquad \text{and} \qquad \nabla\phi = \sum_{i=1}^{m} \mathbf{B}_i^{\phi}\phi_i \tag{32}$$

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

$$\mathbf{B}_i^{\mathbf{u}} = \begin{bmatrix} N_{i,x} & 0 \\ 0 & N_{i,y} \\ N_{i,y} & N_{i,x} \end{bmatrix} \qquad \text{and} \qquad \mathbf{B}_i^{\phi} = \begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} \tag{33}$$

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

$$\begin{aligned} \delta\mathbf{u} &= \sum_{i=1}^{m} \mathbf{N}_i^{\mathbf{u}} \delta\mathbf{u}_i \qquad \text{and} \qquad \delta\phi = \sum_{i=1}^{m} N_i \delta\phi_i \\ \delta\boldsymbol{\epsilon} &= \sum_{i=1}^{m} \mathbf{B}_i^{\delta\mathbf{u}} \mathbf{u}_i \qquad \text{and} \qquad \nabla\delta\phi = \sum_{i=1}^{m} \mathbf{B}_i^{\phi} \delta\phi_i \end{aligned} \tag{34}$$

Using the above expressions and due to the fact that (29) 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

$$\mathbf{r}_i^{\mathbf{u}} = \int_\Omega \left[(1-\phi)^2 + k\right] (\mathbf{B}_i^{\mathbf{u}})^T \boldsymbol{\sigma}_{\mathbf{0}} \, \mathrm{d}V - \int_\Omega (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{b} \, \mathrm{d}V - \int_{\partial\Omega_h} (\mathbf{N}_i^{\mathbf{u}})^T \mathbf{h} \, \mathrm{d}A \tag{35}$$

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

$$r_i^{\phi} = \int_\Omega \left\{ -2(1-\phi) N_i \, \psi_0(\boldsymbol{\epsilon}) + f(\overline{\alpha}) G_c \left[\frac{1}{\ell} N_i \, \phi + \ell (\mathbf{B}_i^{\phi})^T \nabla\phi \right] \right\} \mathrm{d}V \tag{36}$$

In order to obtain the solutions for which $\mathbf{r}^{\mathbf{u}} = \mathbf{0}$ and $\mathbf{r}^{\phi} = \mathbf{0}$, a quasi-Newton solution method is employed. The method is a built-in feature in Abaqus and the stiffness matrix is updated in each iteration following the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. The initial guess for the system takes the following form:

$$\begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t+\Delta t} = \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}_{t} - \begin{bmatrix} \mathbf{K}^{\mathbf{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{\phi\phi} \end{bmatrix}_t^{-1} \begin{Bmatrix} \mathbf{r}^{\mathbf{u}} \\ \mathbf{r}^{\phi} \end{Bmatrix}_t \tag{37}$$

in which the tangent stiffness matrices are calculated as

$$\begin{aligned} \mathbf{K}_{ij}^{\mathbf{uu}} &= \frac{\partial \mathbf{r}_i^{\mathbf{u}}}{\partial \mathbf{u}_j} = \int_\Omega \left[(1-\phi)^2 + k\right] (\mathbf{B}_i^{\mathbf{u}})^T \mathbf{C}_{\mathbf{0}} \, \mathbf{B}_j^{\mathbf{u}} \, \mathrm{d}V \\ \mathbf{K}_{ij}^{\phi\phi} &= \frac{\partial r_i^{\phi}}{\partial \phi_j} = \int_\Omega \left\{ \left[2\psi_0(\boldsymbol{\epsilon}) + \frac{f(\overline{\alpha}) G_c}{\ell} \right] N_i N_j + f(\overline{\alpha}) G_c \ell (\mathbf{B}_i^{\phi})^T (\mathbf{B}_j^{\phi}) \right\} \mathrm{d}V \end{aligned} \tag{38}$$

In the case of a regular Newton-Raphson solution strategy, the system matrix will also contain coupling terms. However, as the initial guess for the quasi-Newton Algorithm these must be neglected [21]. It should be noted that above system of equations does not guarantee the irreversibility of the evolution of the crack phase field, i.e.

$$\phi_{t+\Delta t} \geq \phi_t \tag{39}$$

However, such a constraint can be sufficiently enforced by implementing the history variable approach first proposed in [5]. For the purpose of the phase field equations, ψ_0 is replaced by the history variable H defined as:

$$H = \max_{\tau \in [0,t]} \left(\psi_0(t) \right) \tag{40}$$

3.2. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm

Consider the equation system (37). In quasi-Newton methods, in contrast to standard Newton, the stiffness matrix $\mathbf{K}$ is not updated after each iteration. Instead, after a set number of iterations without convergence, an approximation of the stiffness $\tilde{\mathbf{K}}$ is introduced. This approximated stiffness matrix $\tilde{\mathbf{K}}$ satisfies the following:

$$\tilde{\mathbf{K}} \Delta \mathbf{z} = \Delta \mathbf{r} \tag{41}$$

where

$$\mathbf{z} = \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix}$$

and $\Delta \mathbf{z} = \mathbf{z}_{t+\Delta t} - \mathbf{z}_t$. Likewise, $\Delta \mathbf{r} = \mathbf{r}_{t+\Delta t} - \mathbf{r}$. In the BFGS algorithm, the approximated stiffness matrix is updated in the following way:

$$\tilde{\mathbf{K}} = \tilde{\mathbf{K}}_t - \frac{(\tilde{\mathbf{K}}_t \Delta \mathbf{z})(\tilde{\mathbf{K}}_t) \Delta \mathbf{z})^T}{\Delta \mathbf{z} \tilde{\mathbf{K}}_t \Delta \mathbf{z}} + \frac{\Delta \mathbf{r} \Delta \mathbf{r}^T}{\Delta \mathbf{z}^T \Delta \mathbf{r}} \tag{42}$$

Note that, although the non-diagonal coupling terms of the initial stiffness matrix have been dropped, see (37), the approximation (42) couples the displacement and phase fields. Also, if the stiffness matrix is symmetric, the update to the approximate stiffness matrix can instead be written in terms of its inverse [23]:

$$\tilde{\mathbf{K}}^{-1} = \left(\mathbf{I} - \frac{\Delta \mathbf{z} \Delta \mathbf{r}^T}{\Delta \mathbf{z}^T \Delta \mathbf{r}} \right) \tilde{\mathbf{K}}_t^{-1} \left(\mathbf{I} - \frac{\Delta \mathbf{z} \Delta \mathbf{r}^T}{\Delta \mathbf{z}^T \Delta \mathbf{r}} \right)^{-1} + \frac{\Delta \mathbf{z} \Delta \mathbf{z}^T}{\Delta \mathbf{z}^T \Delta \mathbf{r}}, \tag{43}$$

which offers significant computational savings and retains symmetry and positive definiteness, if such was already present. The BFGS algorithm has been implemented in most commercial finite element packages (such as Abaqus), often in conjunction with a line search algorithm.

3.3. 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` for the purpose of history dependent variables. 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)
Crack phase field - ϕ	SVARS(9)
History variable field - H	SVARS(10)
Cummulative history variable - $\overline{\alpha}$	SVARS(11)
Fatigue history variable - α	SVARS(12)

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
Cummulative history variable - $\overline{\alpha}$	SDV10
Fatigue history variable - α	SDV11

Table 2: List of solution dependent state variables.

3.4. 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)

- The quasi-Newton solution method must be chosen when defining the step. The option can be found under Edit Step → Other → Solution Technique.

- The mesh has to be very refined in the expected crack propagation area. As discussed in [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 [24]. 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=6, coordinates=2, var=48
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` (12 per integration point). We have defined the ordering of the DOFs in a way that (37) 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, 0.004, 2.7, 1e-07,1
*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 - numerical conditioning parameter
PROPS(6)	Fatigue flag parameter

Table 3: List of user element properties.

The last parameter controls whether the fatigue extension is active (1) or inactive (0). For monotonic loading, the current version of the fatigue implementation will have an effect on the result and should be disabled for monotonic loading problems.

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.

While the quasi-Newton solution method is chosen during the creation of the original input file, we here include the keyword which should be present in the step definition of an input file to specify the method:

```
    *Solution Technique, type=QUASI-NEWTON
```

Note that the method is not compatible with step types, analyses or elements which preclude symmetric stiffness matrix storage and solution.

3.5. 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. 3. 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. Two loading cases are considered: Monotonic loading with prescribed displacement $u = 0.01$ mm and cyclic loading with amplitude $u = 0.002$ mm.

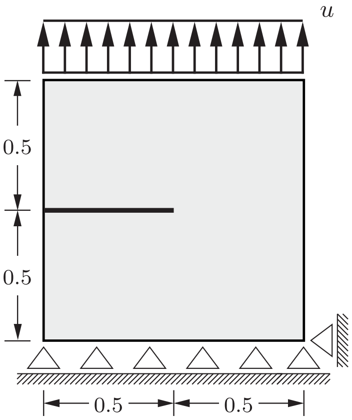

Figure 3: Notched square plate subjected to tension test, geometry and boundary conditions.

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 4887 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=SENTMonotonic user=FatigueQN.f
```

Note that for the fatigue problem, the file `Amplitude.txt` must also be present.

Representative results are shown in Figs. 4 and 5. Specifically, Fig. 4 shows the force versus displacement curve and phase field contour of the monotonic loading problem. The onset of unstable fracture causes a sudden drop in the applied force, as the fracture is captured in a single increment. In the contour plot the red color indicates broken material. For the given mesh, the crack is very diffuse, which can be remedied by refining the mesh and reducing ℓ. In Fig. 5, the crack extension is plotted against the number of cycles. Results for more refined versions of these problems can be found in [25].

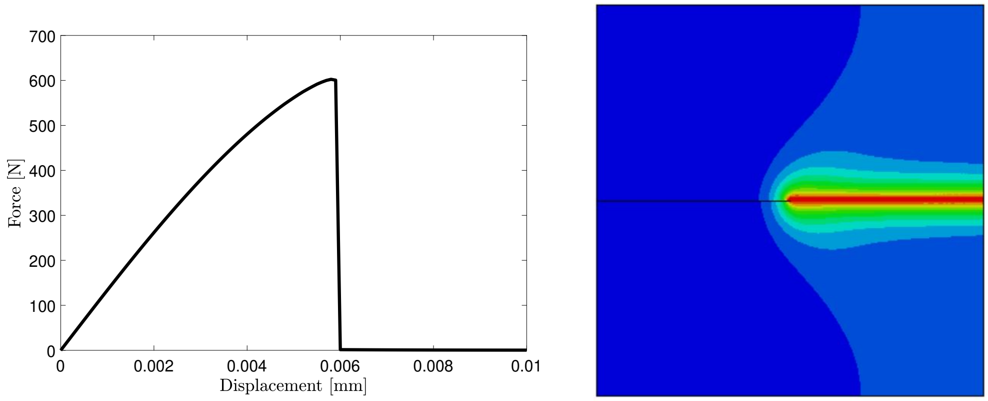

Figure 4: Load-deflection curve for the tension specimen under monotonic loading and contour of the crack phase field.

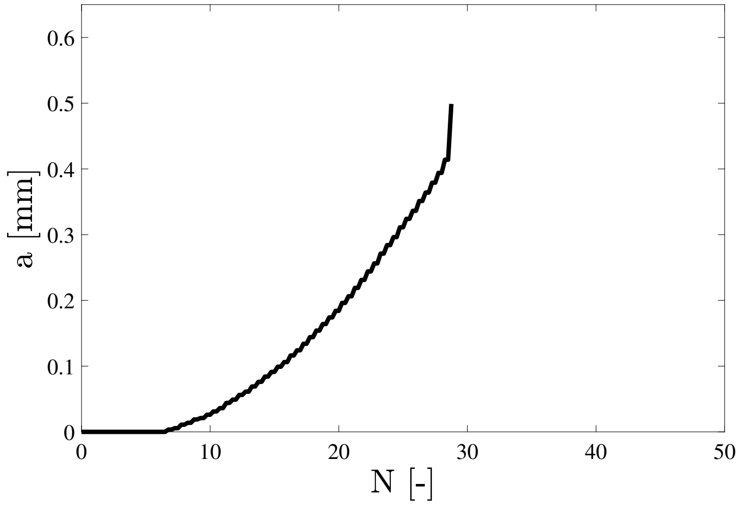

Figure 5: Crack extension vs. cycle number for the cyclically loaded tension specimen.

The input files for these simple boundary value problems are provided in addition to the main subroutine.

4. 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., [26]), the phase field fracture method requires a refined mesh along the potential crack propagation path to resolve the fracture process zone. By use of the quasi-Newton solution algorithm, the phase field fracture model is able to capture unstable crack propagation without the need of control algorithms [27, 28] and independently of the increment size as is the case in the widely used staggered solution scheme [12]. The enormous performance improvements offered by quasi-Newton in comparison to the one-pass staggered iteration scheme are outlined in [13].

5. 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. List of files

SENTMonotonic.inp - Input file for the benchmark problem of a cracked square subjected to tension under monotonic loading.

SENTFatigue.inp - Input file for the benchmark problem of a cracked square subjected to cyclic tension and compression.

FatigueQN.f - UEL Subroutine with the phase field fracture model extended to fatigue.

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] P. Carrara, M. Ambati, R. Alessi, L. De Lorenzis, A framework to model the fatigue behavior of brittle materials based on a variational phase-field approach, Computer Methods in Applied Mechanics and Engineering 361 (2020) 112731.

[11] Y. S. Lo, M. J. Borden, K. Ravi-Chandar, C. M. Landis, A phase-field model for fatigue crack growth, Journal of the Mechanics and Physics of Solids 132 (2019) 103684.

[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] P. K. Kristensen, C. F. Niordson, E. Martínez-Pañeda, A phase field model for elastic-gradient-plastic solids undergoing hydrogen embrittlement, (submitted) (2020).

[14] A. Griffith, The Phenomena of Rupture and Flow in Solids, Philosophical Transactions A, 221 (1920) 163–198.

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

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

[17] G. Bellettini, A. Coscia, Discrete approximation of a free discontinuity problem, Numerical Functional Analysis and Optimization 15 (3-4) (1994) 201–224.

[18] A. Chambolle, An approximation result for special functions with bounded deformation, Journal des Mathematiques Pures et Appliquees 83 (7) (2004) 929–954.

[19] L. Ambrosio, V. M. Tortorelli, Approximation of functionals depending on jumps by elliptic functionals via gamma-convergence, Communications on Pure and Applied Mathematics 43 (1990) (1991) 999–1036.

[20] T. K. Mandal, V. P. Nguyen, J. Y. Wu, Length scale and mesh bias sensitivity of phase-field models for brittle and cohesive fracture, Engineering Fracture Mechanics 217 (2019) 106532.

[21] J.-Y. Wu, Y. Huang, V. P. Nguyen, On the BFGS monolithic algorithm for the unified phase field damage theory, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112704.

[22] K. Pham, H. Amor, J. J. Marigo, C. Maurini, Gradient damage models and their use to approximate brittle fracture, International Journal of Damage Mechanics 20 (4) (2011) 618–652.

[23] H. Matthies, G. Strang, The solution of nonlinear finite element equations, International Journal for Numerical Methods in Engineering 14 (1979) 1613–1626.

[24] 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.

[25] P. K. Kristensen, E. Martínez-Pañeda, Phase field fracture modelling using quasi-Newton methods and a new adaptive step scheme, Theoretical and Applied Fracture Mechanics 107 (2020) 102446.

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

[27] 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.

[28] 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.