A simple yet general ABAQUS phase field fracture implementation using a UMAT subroutine

Yousef Navidtehrani[a], Emilio Martínez-Pañeda[b,*]

[a] *Department of Construction and Manufacturing Engineering, University of Oviedo, Gijón 33203, Spain*
[b] *Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, UK*

Abstract

Documentation that accompanies the UMAT subroutine provided to implement the phase field fracture method in ABAQUS, without the need for user elements. If using these codes for research or industrial purposes, please cite the following articles.

For the UMAT version:
Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda. A unified Abaqus implementation of the phase field fracture method using only a user material subroutine. Materials 14(8), 1913 (2021)

For the UMAT+HETVAL version:
Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda. A simple and robust Abaqus implementation of the phase field fracture method. Applications in Engineering Science 6, 100050 (2021)

Contents

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

1. Introduction and list of files

The phase field fracture method has attained notable popularity. Applications include fibre-reinforced composites [1, 2], hydrogen embrittlement [3–5], batteries [6, 7], rock-like materials [8, 9], functionally graded materials [10–12], corrosion [13], fatigue damage [14, 15] and shape memory alloys [16] - see [17, 18] for an overview.

Phase field modelling has provided a robust numerical platform for the simple yet rigorous fracture thermodynamics principles first presented by Griffith [19]. Complex fracture phenomena such as the merging of cracks, nucleation from arbitrary sites and branching can be capture for arbitrary geometries and dimensions. Damage is described by the phase field ϕ, which goes from 0 (intact material) to 1 (fully cracked), evolving in agreement with the balance between the energy stored in the solid and the energy released during the fracture process. For a solid with material toughness G_c and strain energy density ψ, the phase field ϕ balance law is given by [20]:

$$\nabla^2\phi = \frac{\phi}{\ell^2} - \frac{2\,(1-\phi)}{G_c\ell}\psi\,, \tag{1}$$

where ℓ is the phase field length scale, which governs the size of the damaged region and ensures mesh objectivity. The fact that an additional differential equation (1) has to be solved to predict the evolution of damage complicates the implementation of the phase field fracture method in commercial finite element packages. For example, in Abaqus the implementation is generally done using a user element (UEL) subroutine; that is, programming an *ad hoc* finite element and effectively using Abaqus as a solver. This comes at the cost of not being able to exploit most of Abaqus' in-built features and the complication of the meshing and visualisation processes. Here, we provide a way of implementing the phase field fracture method in Abaqus using only a user material (UMAT) subroutine, without the need of using user elements. This removes the need of extra pre and post-processing steps and enables using Abaqus' in-built capabilities. We achieve this integration point level implementation by exploiting the analogy of the phase field evolution equation (1) with the heat transfer problem; under steady-state conditions, the evolution of the temperature T for a material with thermal conductivity k, which is exposed to a heat source r, is given by,

$$k\nabla^2 T = -r \, . \tag{2}$$

The similarity with Eq. (1) is evident. Accordingly, one can consider the temperature to be the phase field, upon making $k = 1$ and suitably defining r. The definition of r can be achieved inside of a UMAT subroutine for Abaqus version 2020 (or newer), while a HETVAL subroutine should be used for Abaqus versions older than 2020. Both options are provided here. Also, our goal is two-fold. On the one side, we want to provide the simplest phase field fracture implementation in Abaqus. On the other side, we also want to provide a general code, that can accommodate the most widely used constitutive choices. Eq. (1) corresponds to the so called *standard* or `AT2` phase field model, but we also implement the `AT1` and phase field-cohesive zone models (`PF-CZM`). Our implementation also covers the main approaches to split the strain energy density, so as to prevent fracture under compression; that is, the spectral split by Miehe *et al.* [21], and the volumetric-deviatoric approach by Amor *et al.* [22], considering both anisotropic and hybrid approaches [23]. The following files are provided:

UMATs.f - simple UMAT subroutine with the standard (`AT2`) phase field model. To be used with `UMATs.inp`. For Abaqus version 2020 or newer.

HETVALs.f - simple UMAT and HETVAL subroutines with the standard (`AT2`) phase field model. To be used with `HETVALs.inp`.

UMATg.f - General UMAT subroutine with the `AT1`, `AT2` and `PF-CZM` phase field models, including also multiple strain energy splits. To be used with `UMATg.inp`. For Abaqus version 2020 or newer.

HETVALg.f - General UMAT and HETVAL subroutines with the `AT1`, `AT2` and `PF-CZM` phase field models, including also multiple strain energy splits. To be used with `HETVALg.inp`.

The remaining part of the documentation describes: (i) the details of the ABAQUS implementation (Section 2), (ii) the usage instructions (Section 3), and (iii) a representative numerical examples (Section 4). The underlying theory is presented in Appendix A, the heat transfer analogy is described in Appendix B and a plug-in to facilitate further the development of the models is presented in Appendix C. More details can be found in our papers [24, 25].

2. ABAQUS implementation

As, described in Appendix B, one can exploit the analogy between the heat transfer problem and phase field fracture. Thus, the temperature T becomes the phase field ϕ, and will accordingly vary between 0 and 1. A user material (UMAT) subroutine should be used to degrade the stiffness and the stress with the phase field, and to define the heat flux r and its derivative with respect to the temperature (phase field): $\partial r/\partial \phi$. The definition of r and $\partial r/\partial \phi$ should be done in a HETVAL subroutine for Abaqus versions older than 2020.

The procedure is as follows. For a given element, Abaqus provides to the integration point-level subroutines the values of strain and phase field (temperature), as interpolated from the nodal solutions. Within each integration point loop, the user material subroutine (UMAT) is called first. Inside of the UMAT, the material Jacobian $\boldsymbol{C}_0$ and the Cauchy stress $\boldsymbol{\sigma}$ can be readily computed from the strain tensor. The current value of the phase field (temperature) is then used to account for the damage degradation of these two quantities. The strain energy density can be stored in so-called solution dependent state variables (SDVs), enabling to enforce the irreversibility condition (A.11). In the UMAT-only versions (files `UMATs.f` and `UMATg.f`), the internal heat flux r, Eq. (B.4), and its derivative with respect to the temperature (phase field) $\partial r/\partial \phi$, Eq. (B.5), are then defined as the volumetric heat generation (variable `rpl`) and its derivative with respect to the temperature (variable `drpldt`). In the UMAT+HETVAL versions (files `HETVALs.f` and `HETVALg.f`), the definition of r and $\partial r/\partial \phi$ is done in the heat flux (HETVAL) subroutine. The updated value of the SDVs is transferred to the HETVAL subroutine; this is used to transfer the current value of the history field $\mathcal{H}$, without the need for external Fortran modules. The process is repeated for every integration point, enabling Abaqus to externally build the element stiffness matrices and residuals and assembling the global system of equations.

The implementation accommodates both *monolithic* and *staggered* schemes, enabling convergence even in computationally demanding problems. We choose not to define the non-diagonal, coupling terms of the displacement-phase field stiffness matrix; i.e. $\boldsymbol{K}_{\boldsymbol{u}\phi} = \boldsymbol{K}_{\phi\boldsymbol{u}} = \boldsymbol{0}$. This makes the stiffness matrix symmetric. By default, Abaqus assumes a non-symmetric system for coupled displacement-temperature analyses but one can configure the solver to deal with a symmetric system by using the separated solution technique. The current values of the phase field (temperature) and displacement solutions are provided to the subroutine, so they can used to update the relevant variables ($\boldsymbol{C}_0$, $\boldsymbol{\sigma}$, r and $\partial r/\partial \phi$), such that the deformation and fracture problems are solved in a simultaneous (monolithic) manner. Conversely, one can use solution dependent state variables (SDVs) to store and use the history field of the previous increment $\mathcal{H}_t$, effectively freezing its value during the iterative procedure taking place for the current load increment. This is known as a single-pass staggered solution scheme. While single-pass staggered schemes are very robust, unconditional stability no longer holds and one should conduct a sensitivity analysis to ensure that the load increments employed are sufficiently small. Robustness and unconditional stability can be achieved by using quasi-Newton methods [15, 26], but such option is not currently available in Abaqus for coupled temperature-displacement analyses. Independently of the solution scheme, it is known that phase field fracture analyses can achieve convergence after a large number of iterations [15, 27]. Thus, the solution controls are modified to enable this, as discussed below. It should also be noted that parallel calculations using versions of Abaqus older than 2016 only execute the solver in parallel (if the separated solution technique is used).

3. Usage instructions

The same process as for a standard Abaqus model can be followed, with a few extra steps, which a are described below.

1. The material must be defined as a user material (`General - User Material`) with the following attributes:
   (a) To avoid editing the Fortran file, the mechanical and fracture properties are provided as mechanical constants in the user material definition. The simple/standard implementation (files `UMATs.f` and `HETVALs.f`) requires the definition of 5 mechanical constants (`PROPS`), while the generalised implementation (files `UMATg.f` and `HETVALg.f`) requires the definition of 9 constants. These are described in Table 1. The first 5 `PROPS` are common to both the simple and the generalised implementations. The list includes material properties (E, ν, ℓ, G_c, f_t) and solution flag variables. The latter group includes: (i) a flag variable to determine the solution scheme (monolithic vs staggered); (ii) a flag to determine the constitutive model employed, including `AT1` [28], `AT2` [20] and `PF-CZM` (with both linear and exponential softening laws) [29]; (iii) a flag to choose the strain energy split scheme, including the volumetric-deviatoric by Amor *et al.* [22] and the spectral by Miehe *et al.* [21]; and (iv) a flag to decide if the hybrid [23] or the anisotropic [21] splits are used (i.e., whether the split is also applied or not to the balance of linear momentum).

Parameters	Mechanical constants
Young's modulus - E	`PROPS(1)`
Poisson's ratio - ν	`PROPS(2)`
Phase field length scale - ℓ	`PROPS(3)`
Toughness - G_c	`PROPS(4)`
Solution scheme (0 - monolithic, 1 - staggered)	`PROPS(5)`
Model (0: `AT2`; 1: `AT2`; 2: `PF-CZM` [linear]; 3: `PF-CZM` [exp])	`PROPS(6)`
Split (0: No split; 1: Amor *et al.* [22]; 2: Miehe *et al.* [21])	`PROPS(7)`
Split solution scheme (1: hybrid; 2: anisotropic)	`PROPS(8)`
Tensile strength - f_t (only relevant for `PF-CZM`)	`PROPS(9)`

Table 1: Material parameters and solution flags defined by the user. Only the first 5 mechanical constants have to be defined for the simple `AT2` implementation (files `UMATs.f` and `HETVALs.f`).

(b) Solution-dependent state variables (SDVs) must be defined (`General - Depvar`). The number depends on the Fortran file employed: the UMAT-based implementation requires 1 `Depvar` (for both its simple and generalised implementations; files `UMATs.f` and `UMATg.f`), the simple UMAT+HETVAL (file `HETVALs.f`) requires 4 `Depvar` and the generalised UMAT+HETVAL (file `HETVALg.f`) uses 7 `Depvar`. The goal of these solution-dependent state variables is to store the history field $\mathcal{H}$ and, for the UMAT+HETVAL version, to communicate between subroutines. Thus, the only SDV relevant for visualisation purposes is the first one, which corresponds to $\mathcal{H}$.

(c) The conductivity must be set equal to 1 (`Thermal - Conductivity`).

(d) In the case of using UMAT+HETVAL codes (files `HETVALs.f` and `HETVALg.f`), the option `Heat Generation` has to be activated (`Thermal - Heat Generation`).

2. The analysis Step should be of the type `Coupled temp-displacement`, with the following attributes:

   (a) In the `Basic` tab one should select the response to be `Steady-state`. The transient option can be used to add a viscous regularisation parameter, see Refs. [21, 25].

(b) In the `Incrementation` tab, the option `Automatic` should be used. To use a constant increment size (e.g., as for single-pass staggered approaches), set the `Minimum` increment size equal to the `Maximum` one.

(c) In the `Other` tab, one should select the `Separated` solution technique and, subsequently, define the Equation Solver - Matrix storage as `Symmetric`.

3. As phase field fracture analyses can achieve convergence after a large number of iterations [15, 27], the solution controls must be modified to prevent the solver from stopping when a certain number of iterations has been reached. Specifically, set I_0, I_R, I_P, I_C, I_L and I_G to 5000 (in the Step module: `Other - General Solution Controls - Edit...`).

4. A zero temperature initial condition $T(t=0)=0\,\forall\,\boldsymbol{x}$ should be defined for the Initial Step (`Predefined Field - Other - Temperature`). If an initial crack is to be prescribed with $\phi=1$ (instead of geometrically), then an analogous procedure should be used.

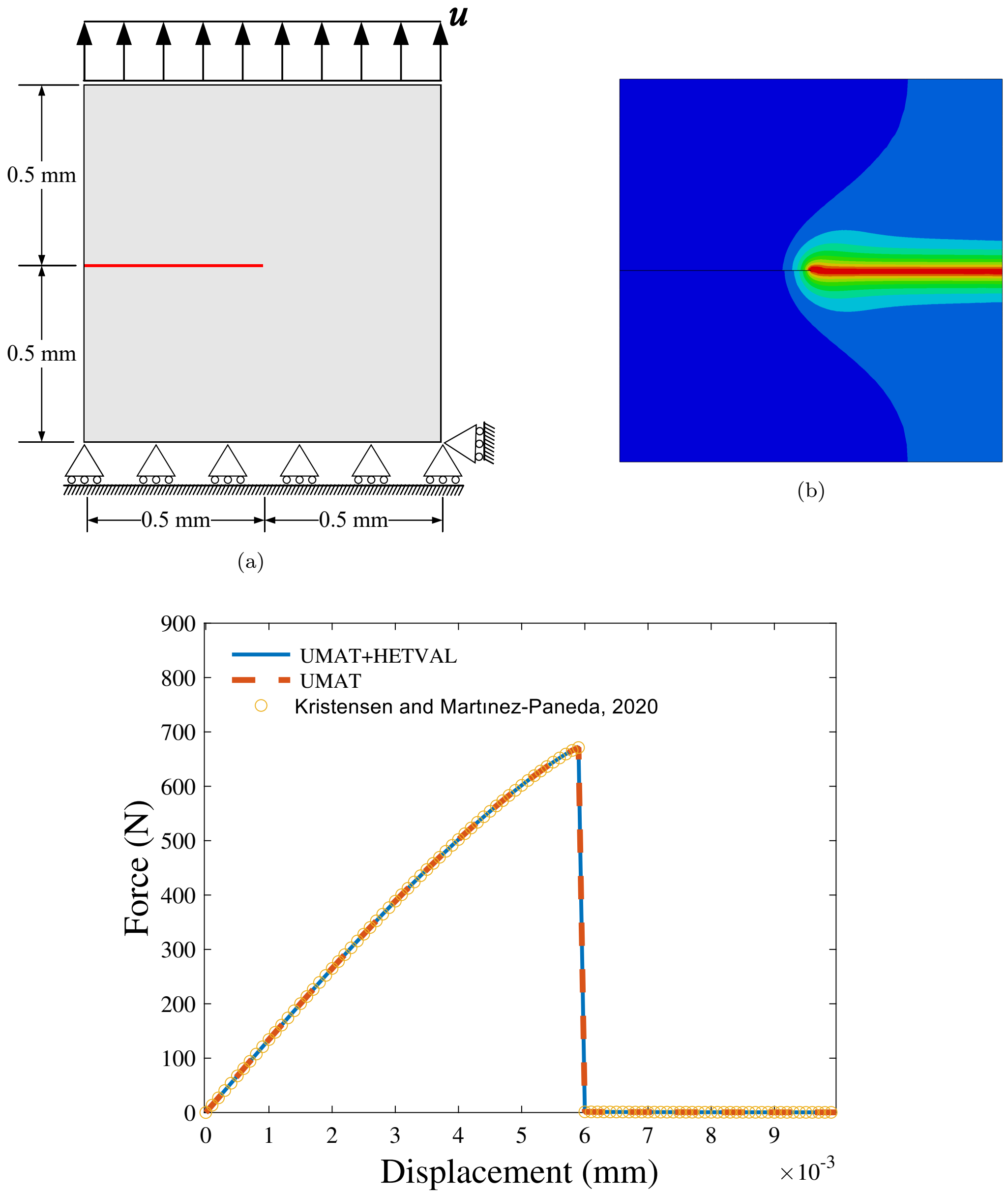
5. In regards to the meshing, the element type should be of the family `Coupled Temperature-Displacement`.

No additional pre-processing or post-processing steps are needed, all actions can be conducted within the Abaqus/CAE graphical user interface and the phase field solution can be visualised by plotting the nodal solution temperature (NT11).

4. Representative example

A simple benchmark is addressed to showcase the use of the subroutine and verify the output - the reader is referred to our papers [24, 25] for further verification case studies and advanced examples. Specifically, we choose to model a paradigmatic benchmark in the phase field fracture community - a square plate with a horizontal crack [15, 21]. The geometry and boundary conditions are shown in Fig. 1a. Young's modulus is chosen to be $E = 210000$ MPa, Poisson's ratio $\nu = 0.3$ and critical energy release rate $G_c = 2.7$ N/mm. The characteristic length scale is $\ell = 0.024$ mm. The load is applied by prescribing a constant total displacement of $u = 0.1$ mm. We discretise the model using linear quadrilateral elements for coupled displacement-thermal analyses, CPE4T in ABAQUS terminology. A total of 8,532 elements are used. The mesh is refined along the expected crack path, such that the characteristic element size is at least five times smaller than the phase field length scale ℓ. For this case study, the monolithic implementation is used and no strain energy decomposition is assumed. The crack path is shown by the contour of phase field in Fig. 1b.

The force versus displacement response predicted is shown in Fig. 1c. The results are shown for both the UMAT and UMAT+HETVAL codes. An excellent agreement is obtained with the result by Kristensen and Martínez-Pañeda [15], which was obtained using a quasi-Newton solution scheme (implemented using a user element subroutine). No convergence issues are observed despite the unstable nature of the fracture process. The input files corresponding to this example are provided with the subroutine files, for illustrative purposes.



(c)

Figure 1: Notched square plate under tension: (a) geometry and boundary conditions, (b) contour of the phase field ϕ after rupture, (c) force versus displacement predictions, for both the UMAT version, the UMAT+HETVAL version and the quasi-Newton results by Kristensen and Martínez-Pañeda [15].

Appendix A. A generalised formulation for phase field fracture

We formulate our generalised formulation, suitable for arbitrary constitutive choices of crack density function and fracture driving force. Consider an elastic body occupying an arbitrary domain $\Omega \subset \mathbb{R}^n$ ($n \in [1,2,3]$), with external boundary $\partial\Omega \subset \mathbb{R}^{n-1}$, on which the outwards unit normal is denoted as $\mathbf{n}$. For a detailed formulation please see Refs. [24, 25].

Appendix A.1. Kinematics

The primal kinematic variables are the displacement field vector $\mathbf{u}$ and the damage phase field ϕ. We restrict our attention to small strains and isothermal conditions. Consequently, the strain tensor $\boldsymbol{\varepsilon}$ reads

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \mathbf{u}^T + \nabla \mathbf{u}\right) . \tag{A.1}$$

The nucleation and growth of cracks are described by using a smooth continuous scalar *phase field* $\phi \in [0;1]$. The phase field describes the degree of damage, being $\phi = 0$ when the material is intact and $\phi = 1$ when the material is fully broken. Since ϕ is smooth and continuous, discrete cracks are represented in a diffuse fashion. The smearing of cracks is controlled by a phase field length scale ℓ. The purpose of this diffuse representation is to introduce the following approximation of the fracture energy over a discontinuous surface Γ:

$$\Phi = \int_\Gamma G_c \, \mathrm{d}S \approx \int_\Omega G_c \gamma(\phi, \nabla\phi) \, \mathrm{d}V, \qquad \text{for } \ell \to 0, \tag{A.2}$$

where γ is the crack surface density functional and G_c is the material toughness [19, 30]. This approximation circumvents the need to track discrete crack surfaces, a well-known challenge in computational fracture mechanics.

Appendix A.2. Principle of virtual work. Balance of forces

Now, we shall derive the balance equations for the coupled deformation-fracture system using the principle of virtual work. The Cauchy stress $\boldsymbol{\sigma}$ is introduced, which is work conjugate to the strains $\boldsymbol{\varepsilon}$. Correspondingly, for an outwards unit normal $\mathbf{n}$ on the boundary $\partial\Omega$ of the solid, a traction $\mathbf{T}$ is defined, which is work conjugate to the displacements $\mathbf{u}$. Regarding fracture, we introduce a scalar stress-like quantity ω, which is work conjugate to the phase field ϕ, and a phase field micro-stress vector $\boldsymbol{\xi}$ that is work conjugate to the gradient of the phase field $\nabla\phi$. The phase field is assumed to be driven solely by the solution to the displacement problem. As a result, no external traction is associated with ϕ. Accordingly, in the absence of body forces, the principle of virtual work is given by:

$$\int_{\Omega} \left\{ \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} + \omega\delta\phi + \boldsymbol{\xi} \cdot \delta\nabla\phi \right\} \mathrm{d}V = \int_{\partial\Omega} \left(\mathbf{T} \cdot \delta\mathbf{u} \right) \mathrm{d}S \tag{A.3}$$

where δ denotes a virtual quantity. This equation must hold for an arbitrary domain Ω and for any kinematically admissible variations of the virtual quantities. Thus, by application of the Gauss divergence theorem, the local force balances are given by:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= 0 \\ \nabla \cdot \boldsymbol{\xi} - \omega &= 0 \end{aligned} \qquad \text{in} \;\; \Omega, \tag{A.4}$$

with natural boundary conditions:

$$\begin{aligned} \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{T} \\ \boldsymbol{\xi} \cdot \mathbf{n} &= 0 \end{aligned} \qquad \text{on} \;\; \partial\Omega. \tag{A.5}$$

Appendix A.3. Constitutive theory

The constitutive theory is presented in a generalised fashion, and the `AT1` [28], `AT2` [20] and `PF-CZM` [29, 31] models are then derived as special cases. The total potential energy of the solid reads,

$$W\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right), \, \phi, \, \nabla\phi\right) = \psi\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right), \, g\left(\phi\right)\right) + \varphi\left(\phi, \, \nabla\phi\right) \tag{A.6}$$

where ψ is the elastic strain energy density and φ is the fracture energy density. The former diminishes with increasing damage through the degradation function $g\left(\phi\right)$, which must fulfill the following conditions:

$$g\left(0\right) = 1, \qquad g\left(1\right) = 0, \qquad g'\left(\phi\right) \leq 0 \;\; \text{for} \;\; 0 \leq \phi \leq 1\,. \tag{A.7}$$

We proceed to formulate the fracture energy density as,

$$\varphi\left(\phi, \, \nabla\phi\right) = G_c \gamma(\phi, \nabla\phi) = G_c \frac{1}{4 c_w \ell} \left(w(\phi) + \ell^2 |\nabla\phi|^2 \right) \,. \tag{A.8}$$

where ℓ is the phase field length scale, Also, c_w is a scaling constant, and $w(\phi)$ is the geometric crack function.

Damage is driven by the elastic energy stored in the solid, as characterised by the undamaged elastic strain energy density ψ_0. To prevent cracking under compressive strain states, the driving force for fracture can be decomposed into active ψ_0^+ and inactive ψ_0^- parts. Accordingly, the elastic strain energy density can be defined as,

$$\psi\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right),\, g\left(\phi\right)\right) = \psi^+\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right), \phi\right) + \psi_0^-\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right)\right) = g\left(\phi\right)\psi_0^+\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right)\right) + \psi_0^-\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right)\right) \tag{A.9}$$

Also, damage is an irreversible process: $\dot{\phi} \geq 0$. To enforce irreversibility, a history field variable $\mathcal{H}$ is introduced, which must satisfy the Karush–Kuhn–Tucker (KKT) conditions:

$$\psi_0^+ - \mathcal{H} \leq 0, \qquad \dot{\mathcal{H}} \geq 0, \qquad \dot{\mathcal{H}}(\psi_0^+ - \mathcal{H}) = 0\ . \tag{A.10}$$

Accordingly, for a current time t, over a total time τ, the history field can be defined as,

$$\mathcal{H} = \text{max}_{t\in[0,\tau]}\psi_0^+\left(t\right)\ . \tag{A.11}$$

Consequently, the total potential energy of the solid (A.6) can be re-formulated as,

$$W = g(\phi)\mathcal{H} + \frac{G_c}{4c_w}\left(\frac{1}{\ell}w(\phi) + \ell|\nabla\phi|^2\right) \tag{A.12}$$

Now we proceed to derive, in a generalised fashion, the fracture micro-stress variables ω and $\boldsymbol{\xi}$. The scalar micro-stress ω is given by,

$$\omega = \frac{\partial W}{\partial \phi} = g'(\phi)\mathcal{H} + \frac{G_c}{4c_w\ell}w'(\phi)\,, \tag{A.13}$$

while the phase field micro-stress vector $\boldsymbol{\xi}$ reads,

$$\boldsymbol{\xi} = \frac{\partial W}{\partial \nabla\phi} = \frac{\ell}{2c_w}G_{\text{c}}\nabla\phi\,. \tag{A.14}$$

Inserting these into the phase field balance equation (A.4b), one reaches the following phase field evolution law:

$$\frac{G_c}{2c_w}\left(\frac{w'(\phi)}{2\ell} - \ell\nabla^2\phi\right) + g'(\phi)\mathcal{H} = 0 \tag{A.15}$$

Appendix B. Heat transfer analogy

Consider a solid with thermal conductivity k, specific heat c_p and density ρ. In the presence of a heat source r, the evolution of the temperature field T in time t is given by the following balance law:

$$k\nabla^2 T - \rho c_p \frac{\partial T}{\partial t} = -r \,, \tag{B.1}$$

Under steady-state conditions the $\partial T/\partial t$ term vanishes and Eq. (B.1) is reduced to,

$$k\nabla^2 T = -r \tag{B.2}$$

Now, rearrange the phase field evolution law (A.15) as,

$$\nabla^2 \phi = \frac{g'(\phi)\,\mathcal{H} 2 c_w}{\ell G_c} + \frac{w'(\phi)}{2\ell^2} \,. \tag{B.3}$$

Equations (B.2) and (B.3) are analogous upon considering the temperature to be equivalent to the phase field $T \equiv \phi$, assuming a unit thermal conductivity $k = 1$, and defining the following heat flux due to internal heat generation,

$$r = -\frac{g'(\phi)\,\mathcal{H} 2 c_w}{\ell G_c} - \frac{w'(\phi)}{2\ell^2} \,. \tag{B.4}$$

Finally, we also define the rate of change of heat flux (r) with temperature ($T \equiv \phi$),

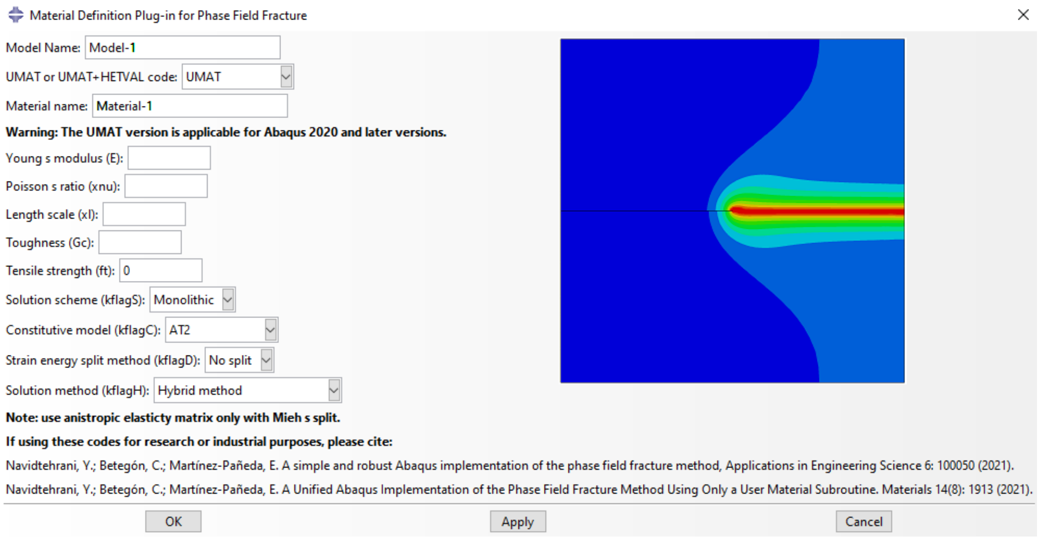
$$\frac{\partial r}{\partial \phi} = -\frac{g''(\phi)\,\mathcal{H} 2 c_w}{\ell G_c} - \frac{w''(\phi)}{2\ell^2} \,, \tag{B.5}$$

as required for the computation of the Jacobian matrix.

Appendix C. PFF_Material Plug-in

A plug-in is provided to facilitate the definition of the material properties for the extended/generalised versions of the UMAT and UMAT+HETVAL codes, aiming to simplify the process. With this plug-in, one can skip Step 1 of the usage instructions (Section 3). Thus, it constitutes the first step in the development of the model; the first action to take after opening Abaqus/CAE. The model and material names are defined as ABAQUS default name unless user wants to change them. The user shall make a decision on the choice of the UMAT or UMAT+HETVAL version, with the default being the UMAT one. A window like that shown in Fig. C.2 will show up, asking the user to introduce Young's modulus, Poisson's ratio, the phase field length scale, and the toughness. The tensile strength should also be defined if using the `PF-CZM` model. All the available options are shown in Fig. C.2.

To install the plugin, the PFF_Material folder must be placed in Abaqus' plugin folder (typically located in C:_SIMULIA_CAE_plugins).



Material Definition Plug-in for Phase Field Fracture

Model Name: Model-1

UMAT or UMAT+HETVAL code: UMAT

Material name: Material-1

Warning: The UMAT version is applicable for Abaqus 2020 and later versions.

Young s modulus (E):

Poisson s ratio (xnu):

Length scale (xl):

Toughness (Gc):

Tensile strength (ft): 0

Solution scheme (kflagS): Monolithic

Constitutive model (kflagC): AT2

Strain energy split method (kflagD): No split

Solution method (kflagH): Hybrid method

Note: use anistropic elasticty matrix only with Mieh s split.

If using these codes for research or industrial purposes, please cite:

Navidtehrani, Y.; Betegón, C.; Martínez-Pañeda, E. A simple and robust Abaqus implementation of the phase field fracture method, Applications in Engineering Science 6: 100050 (2021).

Navidtehrani, Y.; Betegón, C.; Martínez-Pañeda, E. A Unified Abaqus Implementation of the Phase Field Fracture Method Using Only a User Material Subroutine. Materials 14(8): 1913 (2021).

Figure C.2: Material definition plug-in for the extended version of the UMAT and the UMAT+HETVAL codes.

References

[1] A. Quintanas-Corominas, J. Reinoso, E. Casoni, A. Turon, J. A. Mayugo, A phase field approach to simulate intralaminar and translaminar fracture in long fiber composite materials, Composite Structures 220 (2019) 899–911.

[2] W. Tan, E. Martínez-Pañeda, Phase field predictions of microscopic fracture and R-curve behaviour of fibre-reinforced composites, Composites Science and Technology 202 (2021) 108539.

[3] 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.

[4] P. K. Kristensen, C. F. Niordson, E. Martínez-Pañeda, A phase field model for elastic-gradient-plastic solids undergoing hydrogen embrittlement, Journal of the Mechanics and Physics of Solids 143 (2020) 104093.

[5] P. K. Kristensen, C. F. Niordson, E. Martínez-Pañeda, Applications of phase field fracture in modelling hydrogen assisted failures, Theoretical and Applied Fracture Mechanics 110 (2020) 102837.

[6] C. Miehe, H. Dal, L.-M. Schanzel, A. Raina, A phase-field model for chemo-mechanical induced fracture in lithium-ion battery electrode particles, International Journal for Numerical Methods in Engineering 106 (2016) 683–711.

[7] M. Klinsmann, D. Rosato, M. Kamlah, R. M. McMeeking, Modeling crack growth during Li insertion in storage particles using a fracture phase field approach, Journal of the Mechanics and Physics of Solids 92 (2016) 313–344.

[8] S. Zhou, X. Zhuang, T. Rabczuk, Phase field modeling of brittle compressive-shear fractures in rock-like materials: A new driving force and a hybrid formulation, Computer Methods in Applied Mechanics and Engineering 355 (2019) 729–752.

[9] L. Schuler, A. G. Ilgen, P. Newell, Chemo-mechanical phase-field modeling of dissolution-assisted fracture, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112838.

[10] Hirshikesh, S. Natarajan, R. K. Annabattula, E. Martínez-Pañeda, Phase field modelling of crack propagation in functionally graded materials, Composites Part B: Engineering 169 (2019) 239–248.

[11] Hirshikesh, E. Martínez-Pañeda, S. Natarajan, Adaptive phase field modelling of crack propagation in orthotropic functionally graded materials, Defence Technology 17 (2021) 185–195.

[12] P. K. A. V. Kumar, A. Dean, J. Reinoso, P. Lenarda, M. Paggi, Phase field modeling of fracture in Functionally Graded Materials : G - convergence and mechanical insight on the effect of grading, Thin-Walled Structures 159 (2021) 107234.

[13] C. Cui, R. Ma, E. Martínez-Pañeda, A phase field formulation for dissolution-driven stress corrosion cracking, Journal of the Mechanics and Physics of Solids 147 (2021) 104254.

[14] 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.

[15] 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.

[16] M. Simoes, E. Martínez-Pañeda, Phase field modelling of fracture and fatigue in Shape Memory Alloys, Computer Methods in Applied Mechanics and Engineering 373 (2021) 113504.

[17] J.-Y. Wu, V. P. Nguyen, C. T. Nguyen, D. Sutula, S. Sinaie, S. Bordas, Phase-field modelling of fracture, Advances in Applied Mechanics 53 (2020) 1–183.

[18] P. K. Kristensen, C. F. Niordson, E. Martínez-Pañeda, An assessment of phase field fracture: crack initiation and growth, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences (in press) (2021).

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

[20] 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.

[21] 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.

[22] H. Amor, J. J. Marigo, C. Maurini, Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments, Journal of the Mechanics and Physics of Solids 57 (8) (2009) 1209–1229.

[23] M. Ambati, T. Gerasimov, L. De Lorenzis, A review on phase-field models of brittle fracture and a new fast hybrid formulation, Computational Mechanics 55 (2015) 383–405.

[24] Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda, A Unified Abaqus Implementation of the Phase Field Fracture Method Using Only a User Material Subroutine, Materials 14 (8) (2021) 1913.

[25] Y. Navidtehrani, C. Betegón, E. Martínez-Pañeda, A simple and robust Abaqus implementation of the phase field fracture method, Applications in Engineering Science 6 (2021) 100050.

[26] 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.

[27] T. Gerasimov, L. De Lorenzis, A line search assisted monolithic approach for phase-field computing of brittle fracture, Computer Methods in Applied Mechanics and Engineering 312 (2016) 276–303.

[28] 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.

[29] J.-Y. Wu, A unified phase-field theory for the mechanics of damage and quasi-brittle failure, Journal of the Mechanics and Physics of Solids 103 (2017) 72–99.

[30] G. R. Irwin, Onset of Fast Crack Propagation in High Strength Steel and Aluminum Alloys, in: Sagamore Research Conference Proceedings Vol. 2, 1956, pp. 289–305.

[31] J.-Y. Wu, V. P. Nguyen, A length scale insensitive phase-field damage model for brittle fracture, Journal of the Mechanics and Physics of Solids 119 (2018) 20–42.