

# An Abaqus user element (UEL) subroutine to simulate coupled deformation-electrical-fracture behaviour in CNT-based composites

Leonel Quinteros<sup>a,\*</sup>, Enrique García-Macías<sup>a,b</sup>, Emilio Martínez-Pañeda<sup>a</sup>

<sup>a</sup>*Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, UK*

<sup>b</sup>*Department of Structural Mechanics and Hydraulic Engineering, University of Granada, Av. Fuentenueva sn 18002, Granada, Spain*

---

## Abstract

Documentation that accompanies an ABAQUS a user element (UEL) subroutine for implementing the coupled deformation - electrical - phase field fracture scheme proposed by Quinteros et al. [1]. Python scripts are also included to automatise the process and take care of the homogenization step. If using this code for research or industrial purposes, please cite:

Quinteros, L., García-Macías, E., & Martínez-Pañeda, E. (2023). Electromechanical phase-field fracture modelling of piezoresistive CNT-based composites. *Computer Methods in Applied Mechanics and Engineering*, 407, 115941.

The files can be downloaded at [www.imperial.ac.uk/mechanics-materials/codes](http://www.imperial.ac.uk/mechanics-materials/codes) and [www.github.com/L-Quinteros](http://www.github.com/L-Quinteros).

*Keywords:* Abaqus, Carbon nanotubes (CNTs), Finite element analysis, phase field, Piezoresistivity, Smart materials, Fracture

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Numerical model</b>	<b>3</b>
2.1	Governing equations . . . . .	3
2.2	Constitutive equations . . . . .	4
2.2.1	Mechanical deformation . . . . .	4

---

\*Corresponding author.

*Email address:* [l.quinteros-palominos20@imperial.ac.uk](mailto:l.quinteros-palominos20@imperial.ac.uk) (Leonel Quinteros )

2.2.2	Electrical conductivity . . . . .	4
2.2.3	phase field fracture . . . . .	5
2.3	Degradation functions . . . . .	6
2.4	FE implementation . . . . .	7
<b>3</b>	<b>ABAQUS peculiarities and usage instructions</b>	<b>9</b>
3.1	Running a case study . . . . .	9
3.2	Representative results . . . . .	10
<b>4</b>	<b>Summary of included files</b>	<b>11</b>
4.1	User element subroutine . . . . .	12
4.2	Studycases.py . . . . .	12
4.3	Fracture_energy.py . . . . .	13
4.4	MfhFunctions.py . . . . .	14
4.5	Studies.py . . . . .	15
4.6	Case1.inp . . . . .	16
4.7	Case5.inp . . . . .	16
<b>5</b>	<b>Conclusions</b>	<b>16</b>
<b>Appendix A</b>	<b>List of files</b>	<b>17</b>

## 1. Introduction

Piezoresistive carbon nanotube (CNT)-based composites have gained significant attention due to their excellent mechanical and electrical properties [2, 3]. An accurate prediction of the electromechanical behaviour of these composites often requires the use of both homogenization techniques and finite element analysis. This is addressed in this documentation and the accompanying files, which include: (i) Python scripts for the homogenization step (determining effective properties), and (ii) an Abaqus user element (UEL) subroutine to simulate coupled electrical, deformation and (phase field) fracture behaviour. The documentation begins with an introduction to the phase field fracture method and the piezoresistive phase field formulation presented by Quinteros et al. [1]. The focus is on piezoresistive behaviour and the interplay between electric potential and

cracking. Subsequently, Section 3 provides usage instructions while Section 4 describes the files provided and their function.

## 2. Numerical model

### 2.1. Governing equations

Let us consider a solid domain  $\Omega$  with surface  $\partial\Omega$  and normal vector  $\mathbf{n}$ , as illustrated in Fig.1a. A discontinuous surface  $\Gamma$  represents the crack surface. Displacement field and electrical potential are denoted by  $\mathbf{u}$  and  $\varphi$ , respectively. We define an auxiliary phase field variable  $\phi$  ranging from  $\phi = 0$  (intact material) to  $\phi = 1$  (fully broken material). This phase field regularizes the crack surface with a size governed by length scale  $\ell$  [4, 5].

The displacement field's external surface is split into a region of imposed displacements  $\partial\Omega_u$  and a region of traction boundary conditions  $\mathbf{h}$  on  $\partial\Omega_h$  (Fig.1a). An arbitrary crack surface  $\Gamma$  lies within the solid, and a fracture microtraction  $f_\phi$  can be prescribed on  $\partial\Gamma_f$  (Fig.1b). A normal electric current flux  $J_n$  is prescribed in the boundary  $\partial\Omega_{J_n}$ , while the electric potential is prescribed in the boundary  $\partial\Omega_\varphi$  (Fig. 1c). In this context, without body forces, we can formulate the principle of virtual work as:

$$\int_{\Omega} (\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} - \mathbf{J} \cdot \delta\nabla\varphi + \omega \cdot \delta\phi + \zeta \cdot \delta\nabla\phi) dV = \int_{\partial\Omega} (\mathbf{h} \cdot \delta\mathbf{u} + J_n \delta\varphi + f_\phi \delta\phi) dS, \quad (1)$$

Here, the operator  $\delta$  represents first-order variations,  $\boldsymbol{\sigma}$  is the Cauchy stress tensor,  $\mathbf{J}$  denotes the electrical current flow, and  $\omega$  and  $\zeta$  are the microstress work quantities conjugate to the phase field  $\phi$  and its gradient  $\nabla\phi$ , respectively. By applying Gauss' divergence theorem to the previous expression and utilizing the fundamental lemma of the calculus of variations, we obtain the balance of local forces, expressed as:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= \mathbf{0}, \\ \nabla \cdot \mathbf{J} &= 0 \quad \text{in} \quad \Omega, \\ \nabla \cdot \zeta - \omega &= 0, \end{aligned} \quad (2)$$

with the natural boundary conditions,

$$\begin{aligned} \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{h} \quad \text{on} \quad \partial\Omega_h, \\ -\mathbf{J} \cdot \mathbf{n} &= J_n \quad \text{on} \quad \partial\Omega_{J_n}, \\ \zeta \cdot \mathbf{n} &= f_\phi \quad \text{on} \quad \partial\Omega_f. \end{aligned} \quad (3)$$

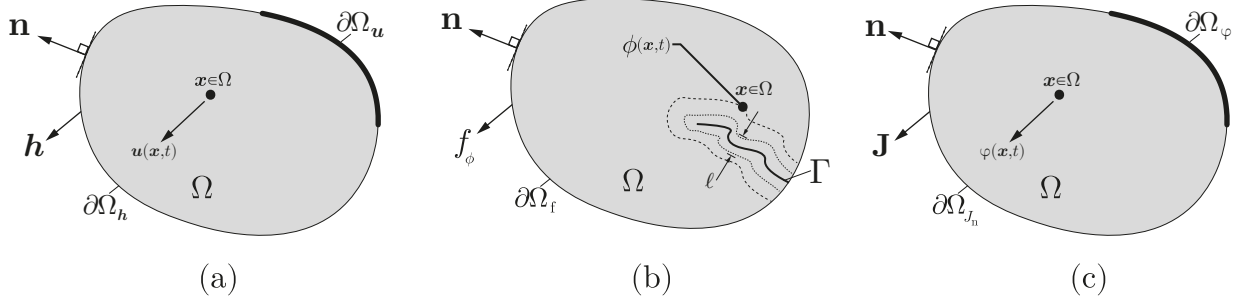


Figure 1: Schematic representation of the three-field boundary value problem: (a) deformation, (b) phase field, and (c) current conservation.

## 2.2. Constitutive equations

The deformation-electrical-fracture couplings involve three key effects. Firstly, the piezoresistivity effect causes mechanical strains  $\boldsymbol{\varepsilon}$  to influence the electrical field  $\mathbf{E}$ . Secondly, mechanical straining increases the stored energy  $\psi_0$  (strain energy density), promoting crack nucleation and growth, which in turn affects the phase field  $\phi$ . Lastly, the presence of cracks influences the electric conductivity, as represented by the phase field degradation of the current flux using an *ad hoc* degradation function. Further constitutive choices are detailed below.

### 2.2.1. Mechanical deformation

The strain field under the assumption of small displacements is expressed as:

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u}^T + \nabla \mathbf{u}), \quad (4)$$

and, assuming a linear elastic relationship between the strains and the undamaged stress tensor  $\boldsymbol{\sigma}_0$ , the mechanical behaviour of the solid is given by

$$\boldsymbol{\sigma} = h_1(\phi) \boldsymbol{\sigma}_0 = h_1(\phi) \mathbf{C} : \boldsymbol{\varepsilon}, \quad (5)$$

where  $\mathbf{C}$  is the linear elastic stiffness tensor, and  $h_1(\phi)$  is a degradation function that relates the phase field variable with the material stiffness.

### 2.2.2. Electrical conductivity

The relation between the electric field  $\mathbf{E}$  and the electric potential  $\varphi$  is given by:

$$\mathbf{E} = -\nabla \varphi, \quad (6)$$

while the constitutive equation is given by the linear relation between the conductivity  $\sigma_{eff}(\epsilon)$ , which is the inverse of the electrical resistivity  $\rho_{eff}$ , i.e.  $\sigma_{eff}(\epsilon) = \rho_{eff}^{-1}$ , and the electric current  $\mathbf{J}$ , which is given by:

$$\mathbf{J} = h_2(\phi)\sigma_{eff}(\epsilon)\mathbf{E}. \quad (7)$$

Here,  $h_2(\phi)$  denotes a second degradation function impacting material conductivity, simulating changes in electrical permeability within cracks. By taking the strong form, Eq. (2)b, and ensuring it holds for any admissible  $\delta\varphi$ , we can obtain the weak form of the electrical problem. Applying the divergence theorem and incorporating the constitutive definitions from Eq. (6) and Eq. (7), we arrive at the following expression:

$$\int_{\Omega} (\delta\nabla\varphi) h_2(\phi)\sigma_{eff}(\epsilon)\nabla\varphi \, dV = \int_{\partial\Omega_{J_n}} J_n \, dS. \quad (8)$$

It is worth noting that the degradation function  $h_2$  can modulate sudden changes in electrical conductivity. Thus, while phase field damage will result in a loss of stiffness and thus high strains, this will not result in a high electric current.

### 2.2.3. phase field fracture

The phase field fracture model forecasts crack evolution as an exchange between stored and fracture energies, based on the thermodynamical balance initially introduced by Griffith [6, 7]. For a cracked solid with strain energy  $\Psi(\epsilon)$  under prescribed displacement, Griffith's energy balance can be represented as the variation in the total potential energy  $\mathcal{E}$  of the solid due to an incremental increase in crack area  $dA$ :

$$\frac{d\mathcal{E}}{dA} = \frac{d\Psi(\epsilon)}{dA} + \frac{dW_c}{dA} = 0, \quad (9)$$

where  $W_c$  is the work required to create new surfaces, with the fracture resistance of the solid (or material toughness) being given by  $G_c = dW_c/dA$ . Equation (9) can be formulated in a variational form as:

$$\mathcal{E} = \int_{\Omega} \psi(\epsilon) \, dV + \int_{\Gamma} G_c \, d\Gamma, \quad (10)$$

where  $\psi$  is the strain energy density of the solid, such that  $\Psi = \int \psi dV$ . Then, to make the minimisation of (10) computationally tractable, the phase field paradigm is introduced, whereby an auxiliary variable  $\phi$  is used to smear an otherwise discrete interface and track the evolution of that interface. Accordingly, a regularised functional can be formulated as:

$$\mathcal{E}_{\ell} = \int_{\Omega} \left[ h_1(\phi)\psi_0(\epsilon) + G_c \left( \frac{\phi^2}{2\ell} + \frac{\ell}{2} |\nabla\phi|^2 \right) \right] \, dV. \quad (11)$$

where  $\psi_0$  denotes the strain energy density of the undamaged material, which for an elastic solid reads:

$$\psi_0 = \frac{1}{2} \boldsymbol{\varepsilon}^T : \mathbf{C} : \boldsymbol{\varepsilon}. \quad (12)$$

In this study, the regularizing term multiplying  $G_c$  in Eq. (11) is chosen in accordance with the AT2 phase field model [4]. For piezoresistive materials, note that the electrical field does not affect the phase field equation, unlike piezoelectric materials [8]. Following thermodynamically consistent criteria [9], we can derive the phase field constitutive equations. The total potential energy of the solid is then given by the sum of the stored and fracture energy densities:

$$\mathcal{W}(\boldsymbol{\varepsilon}, \phi, \nabla \phi) = h_1(\phi) \psi_0(\boldsymbol{\varepsilon}) + G_c \left( \frac{1}{2\ell} \phi^2 + \frac{\ell}{2} |\nabla \phi|^2 \right). \quad (13)$$

The scalar microstress  $\omega$  and the vector microstress  $\zeta$  are then derived from the total potential energy as

$$\omega = \frac{\partial \mathcal{W}}{\partial \phi} = \frac{\partial h_1}{\partial \phi} \psi_0 + G_c \frac{\phi}{\ell}, \quad \text{and} \quad \zeta = \frac{\partial \mathcal{W}}{\partial \nabla \phi} = G_c \ell \nabla \phi. \quad (14)$$

### 2.3. Degradation functions

It remains to define the degradation functions  $h_1(\phi)$  and  $h_2(\phi)$  introduced in Eqs. (5) and (7), respectively. The former describes the loss of stiffness associated with the degradation of material due to damage. For this, we adopt the widely used quadratic function

$$h_1(\phi) = (1 - \phi)^2. \quad (15)$$

On the other side, a degradation function  $h_2(\phi)$  must be defined to account for the variation in electrical permeability due to cracks. To capture the significant increase in local electrical resistivity observed when the material fractures, we propose the following two-parameter exponential function:

$$h_2(\phi, k, n) = \frac{1 - \exp(-k(1 - \phi)^n)}{1 - \exp(-k)}. \quad (16)$$

The degradation function  $h_2(\phi, k, n)$  is shaped by parameters  $k$  and  $n$ , as shown in Fig. 2. These parameters enable the modeling of diverse degradation functions. For example, setting  $n = 6$  and increasing  $k$  simulates more permeable cracks (higher  $h$  values for a given  $\phi$ ). The smoothness of the degradation function is controlled by  $n$ , with sharp decreases in electrical conductivity for low values (e.g.,  $k = 50$  and  $n = 4$ ) and smooth decreases for large values (e.g.,  $k = 50$  and  $n = 8$ ). Finite element predictions will illustrate the influence of various  $k$  and  $n$  choices. Sensible results

and robustness are achieved with  $k = 50$  and  $n = 6$ , which are used throughout this work unless specified otherwise.

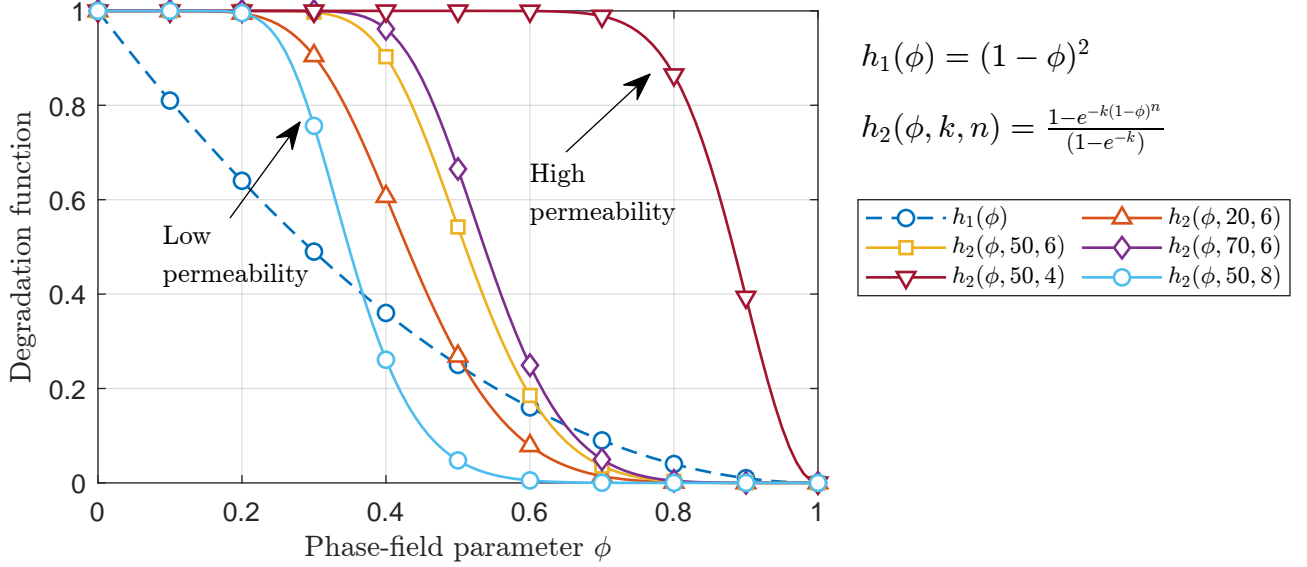


Figure 2: Degradation functions employed to interpolate the phase field,  $h_1(\phi)$ , and the electric conductivity,  $h_2(\phi, k, n)$ , with the latter being dependent on the parameters  $k$  and  $n$ .

It is also important to note that, for numerical reasons, a small regularization parameter  $\epsilon = 10^{-7}$  has been added to both  $h_1(\phi)$  and  $h_2(\phi, k, n)$  to keep the system of equations well-conditioned.

#### 2.4. FE implementation

The finite element (FE) method is chosen to discretise and solve the governing equations provided in Section 2.1. The field variables are the displacement, electric potential and phase field, which are discretised as:

$$\mathbf{u} = \sum_{i=1}^m \mathbf{N}_i \mathbf{u}_i, \quad \varphi = \sum_{i=1}^m N_i \varphi_i, \quad \phi = \sum_{i=1}^m N_i \phi_i, \quad (17)$$

where  $m$  denotes the number of nodes within an element,  $N_i$  are the shape functions, and  $\mathbf{N}_i$  corresponds to diagonal matrices with the nodal shape function  $N_i$  on each component. The strain  $\boldsymbol{\varepsilon}$ , electric field  $\mathbf{E} = -\nabla \varphi$ , and phase field gradient  $\nabla \phi$  are accordingly interpolated as:

$$\boldsymbol{\varepsilon} = \sum_{i=1}^m \mathbf{B}_i^u \mathbf{u}_i, \quad \mathbf{E} = - \sum_{i=1}^m \mathbf{B}_i \varphi_i, \quad \nabla \phi = \sum_{i=1}^m \mathbf{B}_i \phi_i, \quad (18)$$

where  $\mathbf{B}_i$  are the spatial derivatives of the shape function and  $\mathbf{B}_i^u$  denotes the standard strain-displacement matrices. Using the expression for the momentum equilibrium, phase field, and

electrical current conservation from Eq. (2), the weak form corresponding to each of the primary fields can be formulated as:

$$\int_{\Omega} h_1(\phi) \boldsymbol{\sigma}_0 : \delta \boldsymbol{\varepsilon} dV - \int_{\partial \Omega_h} \mathbf{h} \cdot \delta \mathbf{u} dS = 0, \quad (19)$$

$$\int_{\Omega} [h_2(\phi, k, n) (\delta \nabla \varphi) \cdot \boldsymbol{\sigma}_{eff}(\boldsymbol{\varepsilon}) \nabla \varphi] dV - \int_{\partial \Omega_{J_n}} \delta \varphi J_n dS = 0, \quad (20)$$

$$\int_{\Omega} \left[ \frac{\partial h_1}{\partial \phi} \delta \phi \psi_0 + G_c \left( \frac{1}{\ell} \phi \delta \phi + \ell \nabla \phi \cdot \delta \nabla \phi \right) \right] dV - \int_{\partial \Omega_f} f_{\phi} \delta \phi dS = 0. \quad (21)$$

Then, the FE discretization of the residuals can be expressed as:

$$\mathbf{R}_i^u = \int_{\Omega} h_1(\phi) (\mathbf{B}_i^u)^T \boldsymbol{\sigma}_0 dV - \int_{\partial \Omega_h} \mathbf{N}_i^T \mathbf{h} dS, \quad (22)$$

$$\mathbf{R}_i^{\varphi} = \int_{\Omega} [h_2(\phi, k, n) \mathbf{B}_i^T \boldsymbol{\sigma}_{eff}(\boldsymbol{\varepsilon}) \nabla \varphi] dV - \int_{\partial \Omega_{J_n}} N_i^T J_n dS, \quad (23)$$

$$\mathbf{R}_i^{\phi} = \int_{\Omega} \left[ G_c \left( \frac{1}{\ell} N_i \phi + \ell \mathbf{B}_i^T \nabla \phi \right) + \frac{\partial h_1}{\partial \phi} N_i \mathcal{H} \right] dV - \int_{\partial \Omega_f} N_i f_{\phi} dS, \quad (24)$$

in which we adopt the so-called history variable  $\mathcal{H}$  [10] to ensure damage irreversibility, such that  $\mathcal{H} = \max_{t \in [0, t_t]} \psi(t)$  for a time  $t$  within a total time  $t_t$ . Finally, the corresponding stiffness matrices can be stated as:

$$\mathbf{K}_{ij}^u = \frac{\partial \mathbf{R}_i^u}{\partial \mathbf{u}_j} = \int_{\Omega} h_1(\phi) (\mathbf{B}_i^u)^T \mathbf{C} \mathbf{B}_j^u dV, \quad (25)$$

$$\mathbf{K}_{ij}^{\varphi} = \frac{\partial \mathbf{R}_i^{\varphi}}{\partial \varphi_j} = \int_{\Omega} h_2(\phi, k, n) (\mathbf{B}_i)^T \boldsymbol{\sigma}_{eff}(\boldsymbol{\varepsilon}) \mathbf{B}_j dV, \quad (26)$$

$$\mathbf{K}_{ij}^{\phi} = \frac{\partial \mathbf{R}_i^{\phi}}{\partial \phi_j} = \int_{\Omega} \left[ \left( 2\mathcal{H} + \frac{G_c}{\ell} \right) N_i N_j + G_c \ell \mathbf{B}_i^T \mathbf{B}_j \right] dV. \quad (27)$$

And thus the deformation-electrical-damage FE system can be expressed as

$$\begin{Bmatrix} \mathbf{u} \\ \varphi \\ \phi \end{Bmatrix}_{t+\Delta t} = \begin{Bmatrix} \mathbf{u} \\ \varphi \\ \phi \end{Bmatrix}_t - \begin{bmatrix} \mathbf{K}^u & 0 & 0 \\ 0 & \mathbf{K}^{\varphi} & 0 \\ 0 & 0 & \mathbf{K}^{\phi} \end{bmatrix}_t^{-1} \begin{Bmatrix} \mathbf{R}^u \\ \mathbf{R}^{\varphi} \\ \mathbf{R}^{\phi} \end{Bmatrix}_t. \quad (28)$$

The fully coupled system in Eq.(28) is influenced by mechanical deformation affecting both the phase field variable and electrical potential through strain energy density and piezoresistive properties, respectively. Moreover, the phase field degrades the solid's stiffness and electrical conductivity using degradation functions from Section 2.2.3. A monolithic scheme ensures unconditional stability, while robustness and efficiency are achieved using quasi-Newton methods to approximate the stiffness matrix in Eq. (28). The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is employed, as it demonstrates efficient and robust monolithic phase field fracture implementations [11, 12].



### 3. ABAQUS peculiarities and usage instructions

The deformation-electric-phase field model presented is implemented by means of an Abaqus UEL subroutine, which allows for user-defined computation of the element tangent stiffness matrices and the nodal force vectors. We consider isoparametric 3D hexahedron elements with full integration and 5 degrees of freedom per node, i.e.  $u_x$ ,  $u_y$ ,  $u_z$ ,  $\phi$ , and  $\varphi$ .

#### 3.1. Running a case study

This section provides step-by-step instructions on how to run a study case in Abaqus using a Python script.

- Step 1: Open the `StudyCases.py` file in a text editor and locate the `case_number` parameter. Change the value of `case_number` to the desired case number, from 1 to 5.
- Step 2: Modify the electromechanical properties in the same Python file according to the study case requirements.
- Step 3: Open Abaqus and open the toolbar.
- Step 4: Set the working directory to the location of the study case files. This can be done by selecting `File`  $\rightarrow$  `Set Work Directory`, as shown in Fig. 5.
- Step 5: Run the script by executing the command shown in step 2 of Fig. 5. This will launch the study case and execute the Python script with the updated parameters.

A number of quantities are stored as solution-dependent state variables SVARS shown in Table 1.

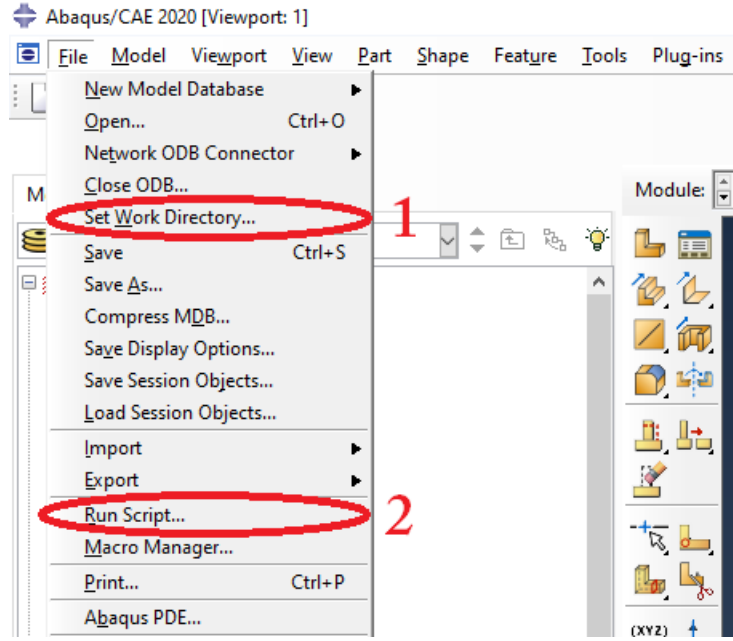


Figure 3: Run a case study from Abaqus using a python file.

Table 1: Simulation parameters

Variable	SVARS numbering
Axial stresses - $\sigma_{11}, \sigma_{22}, \sigma_{33}$	SDV_S11, SDV_S22, SDV_S33
Shear stresses - $\sigma_{12}, \sigma_{23}, \sigma_{13}$	SDV_S12, SDV_S13, SDV_S23
Axial strains - $\sigma_{11}, \sigma_{22}, \sigma_{33}$	SDV_E11, SDV_E22, SDV_E33
Shear strains - $\sigma_{12}, \sigma_{23}, \sigma_{13}$	SDV_E12, SDV_E13, SDV_E23
Crack phase field - $\phi$	SDV_PHI
History variable - $H$	SDV_H
Current flux - $J_1, J_2, J_3$	SDV_JX, SDV_JY, SDV_JZ
Electric potential - $V$	SDV_V

### 3.2. Representative results

We present a benchmark case which corresponds to the `case_number = 5`, where a three-dimensional cylinder with a radius of 2 cm and a length of 5 cm is considered. The sample is fixed at one end while a controlled displacement is applied at the other end, and a potential difference of 10 V is imposed between the two bases of the cylinder. To initiate the defects, five random notches are introduced on the surface of the cylinder by setting the phase field variable equal to

a specified initial condition. The material properties of the cylinder are taken from a [1], with a CNT volume fraction of 1%. A finite element mesh with approximately 320,000 DOFs is used, with the characteristic element length being at least four times smaller than the phase field length.

The results shown in Fig. 5 indicate that the electrical resistance initially exhibits a quasi-linear behaviour dominated by piezoresistance before the defects start to propagate. As the defects propagate, the electrical resistance increases in a non-linear manner according to the degradation function used. Finally, when the crack completely crosses the structure, the current flow between the electrodes is interrupted, and the electrical resistance tends towards infinity. These findings are illustrated in the figure presented in this article.

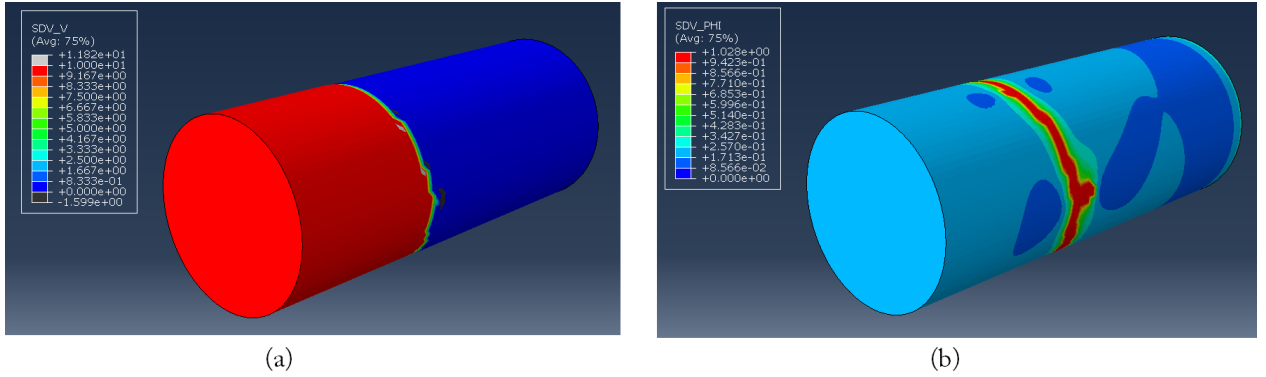


Figure 4: 3D study case contour plot for: (a) Electric potential and (b) phase field variable.

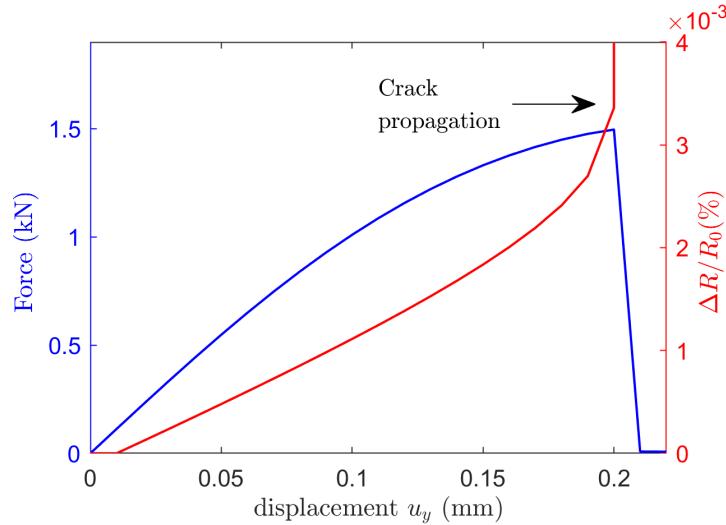


Figure 5: Three-axis plot reporting the relative variation of electrical resistance and load-displacement curves.

#### 4. Summary of included files

The present section provides an explanation of all the files contained within the main folder.

#### 4.1. User element subroutine

The user subroutine, coded in Fortran and named `UEL_piezoresistive_phasefield.for`, represents a 5-degree-of-freedom eight-noded element comprising three degrees of freedom for the mechanical displacement, one degree of freedom for the electric potential, and one degree of freedom for the phase field variable. The subroutine is fully integrated, and the phase field fracture is driven by the linear elastic energy. As a result of the piezoresistive behaviour, the electrical field is modified due to the phase field variable, establishing a relationship between the electrical conductivity and strain.

#### 4.2. *Studycases.py*

The file in question is a Python script that incorporates all the requisite micromechanical variables necessary to obtain the electromechanical properties of CNT-based composites. Additionally, the script executes Cases 1 to 5 with user-defined properties, relying on two other libraries named `MfhFunctions.py` and `Fracture_energy.py` to do so. It also calls the necessary Abaqus libraries to allow the script to be run within the Abaqus environment and requires the `Studies.py` script, which contains all the study cases. Please refer to [github.com/EnriqueGarMac](https://github.com/EnriqueGarMac) for guidance on how to estimate the electrical properties. All the mechanical variables are in Table 2.

Table 2: Micromechanical parameters

Name	Variable	Name	Variable
Volume fraction	<code>fc</code>	Length of MWCNT	<code>Lcnt</code>
Outer diameter of MWCNT	<code>Dcnt</code>	Possion's ratio of MWCNT	<code>NUCnt</code>
Elastic modulus of CNT	<code>ECnt</code>	Maximum CNT orientation angle	<code>ThetaMax</code>
Elastic modulus of epoxy	<code>EMatrix</code>	Minimum CNT orientation angle	<code>ThetaMin</code>
Interphase thickness	<code>intert</code>	Experimental orientation limit angle	<code>Ac</code>
Possion's ratio of epoxy	<code>NuMatrix</code>	Fracture energy of pristine epoxy	<code>G0</code>
Elastic modulus of interphase	<code>EInt</code>	Strength of CNT	<code>SigmaUlt</code>
Interfacial shear strength	<code>TauInt</code>		

The FEM variables, including the applied voltage, displacement, and number of steps, among others are described in Table 3

Table 3: Simulation parameters

Name	Variable	Name	Variable
Applied voltage	Vimp	Name of the file	file_name
Applied displacement	Dimp	Case number	case_number
Increment of steps	Inc	n value	n_value
Notched Angle	NotchedAngle	k value	k_value
mesh size	meshsize1	Length scale parameter	meshsize2

#### 4.3. Fracture\_energy.py

The fracture energy can be estimated using this python class called `FractureEnergy`, which needs:

```

1 class FractureEnergy:
2     def __init__(self, G0, Lcnt, Dcnt, SigmaUlt, TauInt, Ac, mu,
3         Ecnt, ThetaMin, ThetaMax, fc, p=0.5, q=0.5,
4         NInter=100, Ntheta=400):
5         # ----- #
6         # -----Fracture Energy----- #
7         # ----- #
8         self.G0 = G0 # Fracture energy of the pristine
9         self.fc = fc # Cnt volume fraction
10        self.Lcnt = Lcnt # Carbon nanotube length
11        self.Dcnt = Dcnt # Carbon nanotube diameter
12        self.SigmaUlt = SigmaUlt # Ultimate stress
13        self.TauInt = TauInt # Friction shear stress
14        self.Ac = Ac # A coeff
15        self.A = np.power(Dcnt, 2) / 4 # Area
16        self.mu = mu # Snubbing coeff
17        self.Ecnt = Ecnt # Cnt Young's modulus
18        self.ThetaMin = ThetaMin # Theta min integration
19        self.ThetaMax = ThetaMax # Theta min integration
20        self.NInter = NInter # Discretization of the integration of g(theta)
21        self.Ntheta = Ntheta # Discretization of the integration of theta
22        self.p = p # p
23        self.q = q # q

```

Listing 1: Fracture energy class initiation

To estimate the fracture energy you must initiate an object and then apply the `EnergyReleaseRate()` method.

```

1 def EnergyReleaseRate(self):
2     Nint = self.Ntheta
3     ThetaSerie = np.linspace(self.ThetaMin, self.ThetaMax, Nint)
4     Integrand = np.zeros([Nint])
5     for i in range(Nint):
6         theta = ThetaSerie[i]
7         factor = (2 * self.fc) / (self.A * self.Lcnt)
8         Integrand[i] = factor * self.gtheta(theta) * np.cos(theta) * \
9             self.Integranddl(theta)
10    return np.trapz(Integrand, ThetaSerie) + self.G0

```

Listing 2: EnergyReleaseRate method

#### 4.4. *MfhFunctions.py*

The mean field homogenisation is obtained using a python class, which is initiated using the following variables.

```

1 class MFH:
2     def __init__(self, Lcnt, Dcnt, EMatrix, NuMatrix, ECnt,
3                 NuCnt, EInt, NuInt, Intert, fc):
4
5         # -----#
6         # -----Elastic properties-----#
7         # -----#
8         #
9         self.Lcnt = Lcnt
10        self.Dcnt = Dcnt
11        self.EMatrix = EMatrix
12        self.NuMatrix = NuMatrix
13        self.ECnt = ECnt
14        self.NuCnt = NuCnt
15        self.EInt = EInt
16        self.NuInt = NuInt
17        self.Intert = Intert
18        self.fc = fc

```

Listing 3: Mean field homogenisation initiation

Is mandatory to call the `ComputeMechanicalProps()` method to estimate the mechanical properties.

```
1 def ComputeMechanicalProps(self):
2     # Tensors
3     Cm = self.Isotropic(self.EMatrix, self.NuMatrix)
4     Cp = self.Isotropic(self.ECnt, self.NuCnt)
5     # Interphase
6     Kappa = self.Lcnt / self.Dcnt
7     Deq = self.Dcnt * Kappa**(1. / 3.)
8     Ci = self.Isotropic(self.EInt, self.NuInt)
9     Sp = self.EshelbyInt([1, Kappa, 1], self.NuMatrix)
10    Si = self.EshelbyInt([1, Kappa, 1], self.NuMatrix)
11    lambd = self.Intert / Deq
12    fi = self.softinterphase(Kappa, lambd, self.fc)
13    Ceff = self.ellipsoidalinter_random2(Cp, Ci, Cm, Si, Sp, self.fc, fi)
14    E, nu = self.computeEngineeringConstantsSqrt2(Ceff)
15    return E[0], nu[0]
```

Listing 4: `ComputeMechanicalProps` method

#### 4.5. *Studies.py*

This function incorporates several case studies, whereby Case Studies 1 and 5 are invoked through Abaqus `.inp` files and their corresponding Abaqus subroutine. For Cases 2 through 4, the `.inp` files are generated by Abaqus.

```
1 def case_study(
2     Vimp,
3     Dimp,
4     Inc,
5     Eyoung,
6     Nu,
7     Gc,
8     theta,
9     cond,
10    pi11,
11    pi12,
12    pi44,
13    k_value,
```

```

14     n_value ,
15     file_name ,
16     case_study_number ,
17     meshsize ,
18     meshsize2
19 ):

```

Listing 5: Function case\_study

#### 4.6. *Case1.inp*

The input file for case study number 1

#### 4.7. *Case5.inp*

The input file for case study number 5

## 5. Conclusions

The present documentation describes to the best of our efforts the codes developed as part of Ref. [1], which are aimed at providing an improved understanding of the electro-mechanical behaviour of CNT-based composites. The focus is on the fracture behaviour and its interplay with piezoresistivity but its potential applications go far beyond; e.g., the Python scripts provided could be useful for those working in any area of homogenization and the UEL subroutine is suitable for any electro-mechanical fracture study.

## Acknowledgements

L. Quinteros acknowledges financial support from the National Agency for Research and Development (ANID)/ Scholarship Program / DOCTORADO BECAS CHILE/2020 - 72210161. E. García-Macías was supported by the Consejería de Transformación Económica, Conocimiento, Empresas y Universidades de la Junta de Andalucía (Spain) through the research project P18-RT-3128. E. Martínez-Pañeda was supported by an UKRI Future Leaders Fellowship (grant MR/V024124/1).



## Appendix A. List of files

**Case1.inp** - The input file for case study number 1

**Case5.inp** - The input file for case study number 5

**Fracture\_energy.py** - Python script to automatically estimate the fracture energy.

**MfhFunctions.py** - Python script to automatically estimate mechanical properties using mean field homogenisation techniques.

**Studies.py** - Python script to automatically generates the case studies, form 1 to 5.

**StudyCases.py** - Python script that incorporates all the requisite micromechanical variables necessary to obtain the electromechanical properties of CNT-based composites. Additionally, the script executes Cases 1 to 5 with user-defined properties, relying on two other libraries named **MfhFunctions.py** and **Fracture\_energy.py**.

**UEL\_piezoresistive\_phasefield.for** - This user subroutine, coded in Fortran, represents a 5-degree-of-freedom eight-noded element comprising three degrees of freedom for the mechanical displacement, one degree of freedom for the electric potential, and one degree of freedom for the phase field fracture variable.

## References

- [1] L. Quinteros, E. García-Macías, E. Martínez-Pañeda, Electromechanical phase-field fracture modelling of piezoresistive cnt-based composites, *Computer Methods in Applied Mechanics and Engineering* 407 (2023) 115941.
- [2] T. Hassan, A. Salam, A. Khan, S. U. Khan, H. Khanzada, M. Wasim, M. Q. Khan, I. S. Kim, Functional nanocomposites and their potential applications: A review, *Journal of Polymer Research* 28 (2) (2021) 1–22.
- [3] L. Quinteros, E. García-Macías, E. Martínez-Pañeda, Micromechanics-based phase field fracture modelling of CNT composites, *Composites Part B: Engineering* 236 (2022) 109788.
- [4] B. Bourdin, G. Francfort, J.-J. Marigo, Numerical experiments in revisited brittle fracture, *Journal of the Mechanics and Physics of Solids* 48 (4) (2000) 797–826.
- [5] 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* 379 (2203) (2021) 20210021.

- [6] A. A. Griffith, The Phenomena of Rupture and Flow in Solids, *Philosophical Transactions A*, 221 (1920) 163–198.
- [7] G. A. 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.
- [8] C. Miehe, F. Welschinger, M. Hofacker, A phase field model of electromechanical fracture, *Journal of the Mechanics and Physics of Solids* 58 (10) (2010) 1716–1740.
- [9] Z. Khalil, A. Y. Elghazouli, E. Martínez-Pañeda, A generalised phase field model for fatigue crack growth in elastic–plastic solids with an efficient monolithic solver, *Computer Methods in Applied Mechanics and Engineering* 388 (2022) 114286.
- [10] 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) (2010) 2765–2778.
- [11] 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.
- [12] 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.