A control algorithm to overcome convergence problems in cohesive zone model analyses

Emilio Martínez-Pañeda[a,*]

[a] *Department of Engineering, Cambridge University, CB2 1PZ Cambridge, UK*

Abstract

Documentation that accompanies the Abaqus input files with representative examples of the use of a control algorithm to avoid convergence problems in finite element simulations of crack propagation in cohesive interfaces. The model has been largely inspired by the pioneering work by Segurado and LLorca (2004) on particle fracture and interface decohesion in composites. If using this code for research or industrial purposes, please cite their work and the following paper:

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

Keywords:
Cohesive zone model, Finite element analysis, Fracture, Crack growth, Numerical instabilities, Convergence

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

1. Introduction

Cohesive zone models constitute a convenient way of modeling crack initiation and subsequent propagation. Damage is restricted to evolve along the predefined cohesive interface and the micromechanisms of material degradation and failure are embedded into the constitutive law that relates the cohesive traction with the local separation. This traction-separation law that governs material degradation and separation is the pivotal ingredient of cohesive zone formulations. As depicted in Fig. 1, for a given shape of the traction-separation curve, the cohesive response can be fully characterized by two parameters, the cohesive energy Γ_c and the critical cohesive strength T_c. Thus, for the bi-linear law of Fig. 1, the cohesive energy can be expressed as a function of the critical separation δ_c and the critical cohesive strength T_c,

$$\Gamma_c = \frac{1}{2} T_c \delta_c \tag{1}$$

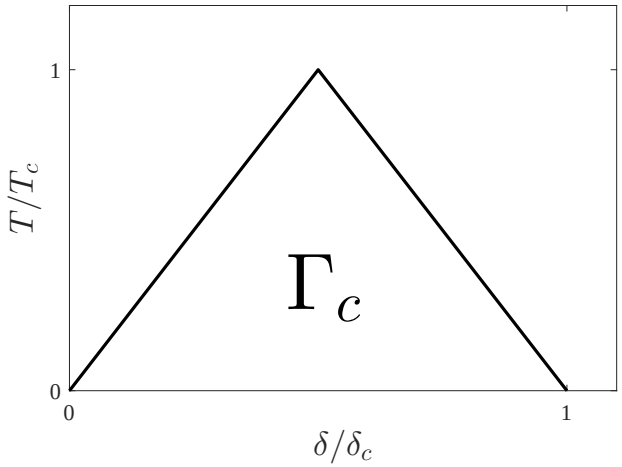

Figure 1: Cohesive bi-linear traction separation law.

The critical cohesive traction T_c and the fracture energy Γ_c can be respectively assimilated to the cohesive bonding strength between atoms and the critical energy release rate G_c. Therefore, cohesive parameters can be directly related to the underlying physics, and cohesive zone analyses have provided important physical insight into the fracture process of metals (Martínez-Pañeda et al., 2017b; Tvergaard and Hutchinson, 1992). Contrary to these successes, cohesive zone models are prone to convergence problems. The softening part of the traction-separation law gives rises to a local stiffness degradation in the corresponding cohesive elements, which triggers elastic snap-back instabilities. Hence, at the point where the stress reaches the peak strength of the interface, quasi-static finite element computations are unable to converge to an equilibrium solution, hindering the modeling of the post-instability behavior. Viscous regularization techniques have been proposed (see, e.g. Gao and Bower, 2004) but convergence problems are still observed and the solution can easily deviate from the equilibrium solution of the original problem if the viscosity coefficient is not sufficiently small. A suitable numerical strategy is here presented to efficiently overcome convergence problems in cohesive crack propagation studies. The control algorithm employed was first presented by Segurado and LLorca (2004) and can be easily incorporated in a commercial finite element package, Abaqus is here used as an example.

2. Control Algorithm

Let us imagine a plate of height H and thickness W with an edge crack a_0, as shown in Fig. 2. We would like to see how the crack propagates in

this plane strain mode I problem by taking advantage of symmetry (i.e., we only model the upper half of the specimen) and by placing cohesive elements ahead of the crack. However, as damage evolves we will eventually reach an instability point where both the reaction forces and the remote vertical displacement that we are prescribing will decrease.

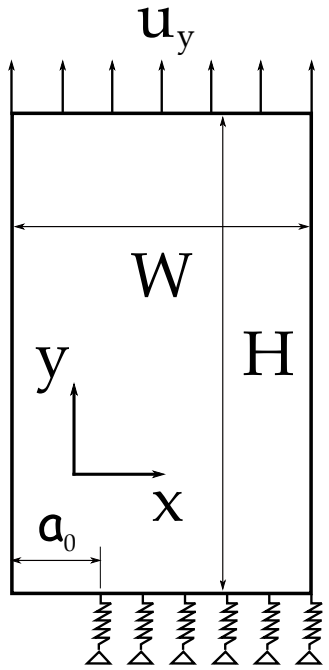

Figure 2: Example geometry: edge cracked plate. Cohesive elements are graphically represented by springs.

Like in the pioneering mixed FE-Rayleigh-Ritz method proposed by Tvergaard (1976), the idea is to capture the simultaneous reduction of the load and the displacement at the remote boundary by finding a variable that increases monotonically during the whole loading history. The control algorithm enables to provide as input the magnitude of this monotonically increasing quantity while obtaining the remote load as an output of the

equilibrium solution. In a crack propagation simulation the vertical displacement at the crack tip increases monotonically. One could choose to control the crack tip node or to control (in addition) a number of nodes ahead of the crack tip. We will focus first on the former, as it is easier to understand, and explain the latter afterwards (which could be more convenient in the case of "local" snap-backs).

2.1. First case: crack tip control

We choose to prescribe the opening displacement at the crack tip and use the control algorithm to obtain as outcome of our finite element model the equilibrium solution. The control algorithm is basically equivalent to adding two new equations to our existing finite element global system of equations. In order to do so we define a *control* node N_c that can be located anywhere, and we relate, (i) the force in this control node to the displacement of the crack tip $f_y^{N_C} = u_y^{N_1}$, and (ii) the displacement in this control node to the force in a representative node of the outer boundary $u_y^{N_C} = f_y^{N_L}$. Here, N_C is the control node, N_1 is the crack tip node and N_L is the node at the remote boundary (all the other nodes at the remote boundary are forced to displace vertically as N_L). In this way we can prescribe in our input file the force of the control node (which is equivalent to prescribing the crack tip displacement) and the displacement at the control node (i.e., the remote load) will be an outcome of the equilibrium solution. The way of defining these relations is by defining two auxiliary elements that will connect the control node with N_L and N_1 as shown in Fig. 3.

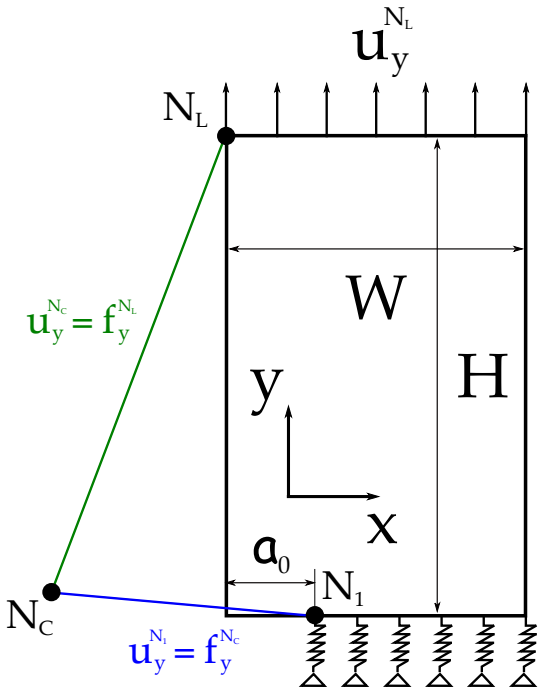

Figure 3: Simplified outline of the control algorithm employed to prescribe the crack tip displacement $u_y^{N_1}$ and obtain the force in the remote boundary $f_y^{N_L}$ from the equilibrium response.

The first auxiliary element connects the vertical displacement of the crack tip node N_1 to the force in the control node by means of the following stiffness matrix,

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u_y^{N_1} \\ u_y^{N_c} \end{pmatrix} = \begin{pmatrix} f_y^{N_1} \\ f_y^{N_c} \end{pmatrix} \tag{2}$$

such that the crack tip opening displacement is linearly related to the vertical force in the control node. The second auxiliary element is defined to equate the displacement in the control node to the vertical load in one of the nodes in the outer boundary N_L, where a remote displacement would otherwise be

prescribed.

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_y^{N_L} \\ u_y^{N_c} \end{pmatrix} = \begin{pmatrix} f_y^{N_L} \\ f_y^{N_c} \end{pmatrix} \tag{3}$$

In that way the crack tip opening is prescribed by imposing a vertical force on the control node, and the displacement at the outer boundary is an outcome of the equilibrium solution.

2.2. Second case: average crack opening control

In some circumstances local snap-backs may arise, where the crack tip node decreases decreases shortly at some point in the loading history. To avoid this problem a more robust approach is to prescribe the sum of the relative openings of all the interface elements. Everything will be identical to the case described before with the exception of the first auxiliary element. The stiffness matrix of the first auxiliary element is defined so as to connect the vertical displacement of the nodes ahead of the crack tip $(N_1, N_2, \ldots, N_n)$ to the force in the control node N_c,

$$\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} \begin{pmatrix} u_y^{N_1} \\ u_y^{N_2} \\ \vdots \\ u_y^{N_c} \end{pmatrix} = \begin{pmatrix} f_y^{N_1} \\ f_y^{N_2} \\ \vdots \\ f_y^{N_c} \end{pmatrix} \tag{4}$$

implying $f_y^{N_C} = u_y^{N_1} + u_y^{N_2} + \cdots + u_y^{N_N}$, such that the average opening displacement is linearly related to the vertical force in the control node. In this way the sum of the relative openings is prescribed by imposing a vertical force on the control node and, as in the previous case, the displacement at the outer boundary is an outcome of the equilibrium solution.

3. Numerical implementation in Abaqus

The control algorithm presented can be easily incorporated into commercial finite element packages, Abaqus' case will be shown here as an example. In order to establish a comparison with a standard implementation (i.e., no control algorithm) three models have been developed: (i) the standard case, where the outer displacement is prescribed (Standard.inp), (ii) the case where we use the control algorithm to prescribe the crack tip opening displacement (ControlTip.inp), and (iii) the case where we use the control algorithm to prescribe the sum of the opening displacements of the nodes ahead of the crack (ControlSum.inp).

We take as benchmark a very simple model. We consider the cracked plate outlined in Figure 2, with dimensions $a_0 = 1$ mm, $W = 10a_0$ and $H = 4W$. We also assume linear elastic behavior in the bulk and we employ Abaqus in-built cohesive elements. Using user defined cohesive elements is highly recommended and the user element subroutine employed for this purpose in (del Busto et al., 2017) can be downloaded from www.empaneda.com/codes. However, we will make use of Abaqus in-built cohesive elements to simplify the example as much as possible. A rather coarse mesh is employed, as shown in Fig. 4; our aim is to show the capabilities of the control algorithm but one should bear in mind that an accurate assessment of crack propagation will require a mesh sufficiently fine to properly resolve the cohesive fracture process zone. 900 linear plane strain quadrilateral elements have been employed (CPE4 in Abaqus' notation); 20 elements are used to resolve the extended crack plane (i.e., 20 4-node cohesive elements will be employed).

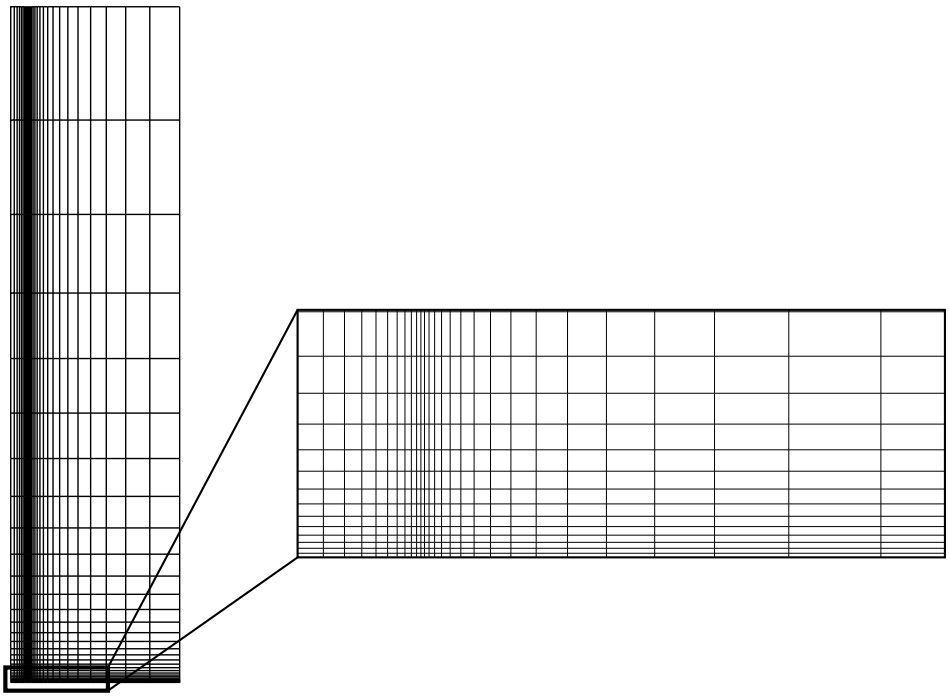

Figure 4: General and detailed representation of the finite element mesh employed.

The next section provides details on how Abaqus zero-thickness in-built cohesive elements have been included in the model, a procedure that it is identical in both the standard and control algorithm cases. Section 3.2 explains in detail how the input file is modified to incorporate the control algorithm (the crack tip case is taken as example), while section 3.3 shows the changes that are needed to prescribe the sum of the openings ahead of the crack.

3.1. Abaqus cohesive elements

To define the cohesive response define first a new Material, where one should introduce the initial linear stiffness of the cohesive law (Mechanical

-> Elasticity -> Elastic and Type: Traction) and the damage initiation criterion (Mechanical -> Damage for Traction-Separation Laws). Within the latter the shape of the cohesive law and the magnitude of the fracture energy can be introduced (Suboptions -> Damage Evolution). The next step involves creating a new Section that will be latter assigned to the cohesive elements (Sections -> Other -> Cohesive). Useful output quantities are SDEG (scalar damage variable) and DMICRT (damage initiation criteria).

To introduce zero-thickness cohesive elements in the mesh it is convenient to operate with the input file. Matlab scripts have been developed for this as part of the Abaqus2Matlab software (Papazafeiropoulos et al., 2017).[1] By means of these scripts one can add the additional nodes, elements (in this example, of the type COH2D4), sets and constraints. Regarding the latter, one should note that since we are taking advantage of symmetry we have to constraint the horizontal displacement of each pair of nodes to be the same. If a cohesive element is placed at a line of symmetry, it will undergo an unsymmetric deformation if simply the displacements of the lower surface are constrained.

In the Standard case (Standard.inp) the load is imposed by prescribing the displacement in the outer boundary. As shown in Section 4 this will lead to convergence problems and the impossibility of tracking the post-instability response.

[1]They may not be part of the official package yet, please contact me at mail@empaneda.com if you want them.

3.2. Crack tip control

We implement the control algorithm in its simplest form (i.e., controlling the crack tip node) by adding the two user elements of Eqs. (2) and (3) in the input file by means of two user defined elements. First, we define the control node, which can be placed anywhere in the model,

```
*NODE,NSET=NodeC
983,0.,0.
```

Next, we define the first auxiliary element following Eq. (2). Following Abaqus' convention the transpose of the stiffness matrix is given and we add a very small number in the diagonal to avoid singularities.

```
*USER ELEMENT,NODES=2,TYPE=U1,LINEAR,UNSYMM
2
*MATRIX,TYPE=STIFFNESS
1.e-10,1
0.0,1.e-10
```

We specify the element as linear, involving 2 nodes, with only the 2nd degree of freedom active, and with an non-symmetric stiffness matrix. This first auxiliary element involves the control node and the node at the crack tip (621 in this example),

```
*ELEMENT,TYPE=U1,ELSET=control
921,621,983
```

Then we define the second auxiliary element, given by Eq. (3). Again, a linear element with 2 nodes, one degree of freedom and a non-symmetric stiffness matrix is defined,

```
*USER ELEMENT,NODES=2,TYPE=U2,LINEAR,UNSYMM
2
*MATRIX,TYPE=STIFFNESS
1.e-10,0.0
1.0,1.e-10
```

This second auxiliary element links the control node with the representative node at the outer edge (961 in this example).

```
*ELEMENT,TYPE=U2,ELSET=control
922,961,983
*UEL PROPERTY,ELSET=control
```

We use sets to identify each node and a constraint equation is defined to make sure that the nodes in the outer boundary displace equally in the vertical direction.

```
*Equation
2
Ntop, 2, 1.
N_L, 2, -1.
```

The global stiffness matrix will no longer by symmetric and one should take this into account in the Step definition,

```
*Step, name=Step-1, inc=10000, unsymm=YES
```

And finally a concentrated force is prescribed in the control node,

```
*CLOAD
N_C,2,0.005
```

This is the load boundary condition of our problem and its magnitude will be equal to the displacement that we wish to prescribe at the crack tip.

3.3. Average crack opening control

The implementation if we want to prescribe the sum of the openings ahead of the crack is very similar to the one outlined in Section 3.2. The only difference lies in the definition of the 1st auxiliary element, in this case an element is listed per node ahead of the crack. E.g., for the example under consideration, with 20 elements ahead of the crack, one should have,

```
*ELEMENT,TYPE=U1,ELSET=control
921,621,983
922,590,983
923,559,983
924,528,983
925,497,983
926,466,983
927,435,983
928,404,983
929,373,983
930,342,983
931,311,983
932,280,983
933,249,983
934,218,983
935,187,983
```

```
936,156,983
937,125,983
938,94,983
939,63,983
940,32,983
941,1,983
```

The force that we prescribe will correspond to sum of opening displacements ahead of the crack (the averaged opening multiplied by the number of nodes).

4. Results

We show the capabilities of this numerical strategy by considering a linear elastic plate of Young's modulus $E = 200000$ MPa and Poisson's ratio $\nu = 0.3$. We choose a bilinear cohesive law, like the one outlined in Fig. 1, and define an initial stiffness of $K = 1000E$. Damage initiates when the critical cohesive traction reaches $T_c = 600$ MPa and the fracture energy equals $\Gamma_c = 2$ MPa mm (since we are taking advantage of symmetry we introduce $\Gamma_c = 1$ MPa mm in the numerical model). The results obtained by running the files Standard.inp and ControlTip.inp are shown in Fig. 5 in terms of the force vs displacement curve. The outcome of the file ControlSum.inp is not shown, as it falls on top of the one obtained for the crack tip opening displacement.

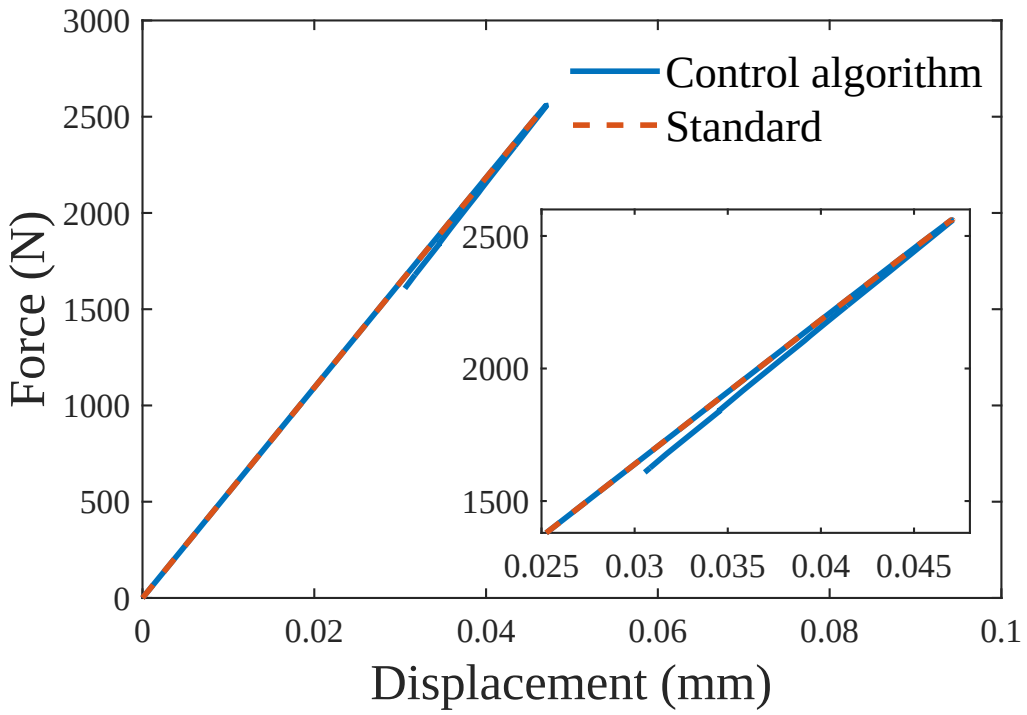

Figure 5: Reaction vertical forces versus remote displacement obtained by the standard and the present approaches.

As shown in the figure, in the standard case, where we prescribe the displacement in the outer boundary, the model crashes shortly after crack initiation as it is unable to capture the simultaneous reduction in load and displacement displayed by the plate. Contrarily, the job with the control algorithm finish successfully and enables to capture the post-instability response.

5. Conclusions

The implementation of the control algorithm employed in (Martínez-Pañeda et al., 2017a) has been described in detail. The numerical strategy presented allows to conduct crack propagation studies with cohesive elements overcoming the numerical convergence problems typically associated with these analyses. The technique builds on the work by (Segurado and LLorca, 2004) on interface decohesion in composites and is far more efficient than other similar approaches (such as the mixed FE-Rayleigh-Ritz method). Moreover, it can be easily incorporated in commercial finite element packages, as shown for the case of Abaqus in this accompanying documentation.

This technique could be used with other numerical methods for crack propagation (X-FEM, phase field, continuum damage models, etc.) and for other boundary value problems where instabilities may arise (e.g., buckling).

6. Acknowledgments

E. Martínez-Pañeda also 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).

References

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

Gao, Y. F., Bower, A. F., 2004. A simple technique for avoiding convergence problems in finite element simulations of crack nucleation and growth on cohesive interfaces. Modelling and Simulation in Materials Science and Engineering 12 (3), 453–463.

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

Martínez-Pañeda, E., Deshpande, V., Niordson, C. F., Fleck, N. A., 2017b. Crack growth resistance in metals. (submitted).

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

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

Tvergaard, V., 1976. Effect of thickness inhomogeneities in internally pressurized elastic-plastic spherical shells. Journal of the Mechanics and Physics of Solids 24 (5), 291–304.

Tvergaard, V., Hutchinson, J. W., 1992. The relation between crack growth resistance and fracture process parameters in elastic-plastic solids. Journal of the Mechanics and Physics of Solids 40 (6), 1377–1397.