Development of a user element in ABAQUS for modelling of cohesive laws in composite structures

Feih, Stefanie

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

RISØ

Risø-R-1501(EN)

Development of a user element in ABAQUS for modelling of cohesive laws in composite structures

Stefanie Feih

Risø National Laboratory
Roskilde
Denmark
January 2005

Author: Stefanie Feih
Title: Development of a user element in ABAQUS for modelling of cohesive laws in composite structures
Department: AFM

Risø-R-1501(EN)
January 2005

Abstract (max. 2000 char.):

The influence of different fibre sizings on the strength and fracture toughness of composites was studied by investigating the characteristics of fibre cross-over bridging in DCB specimens loaded with pure bending moments. These tests result in bridging laws, which are obtained by simultaneous measurements of the crack growth resistance and the end opening of the notch. The advantage of this method is that these bridging laws represent material laws independent of the specimen geometry. However, the adaption of the experimentally determined shape to a numerically valid model shape is not straight forward, and most existing publications consider theoretical and therefore simpler softening shapes. In this article, bridging laws were implemented into an interface element in the UEL user subroutine in the finite element code ABAQUS. Comparison with different experimental data points for crack opening, crack length and crack shape show the sensitivity of these results to the assumed bridging law shape. It is furthermore shown that the numerical predictions can be used to improve the bridging law fit. One shape with one adjustable parameter then fits all experimental data sets.

ISSN 0106-2840
ISBN 87-550-3410-1

Contract no.:

Group's own reg. no.:
16150043-00

Sponsorship:

Cover :

Pages: 52
Tables: 3
Figures: 26
References: 15

Print: Pitney Bowes Management Services Denmark A/S, 2005

Risø National Laboratory
Information Service Department
P.O.Box 49
DK-4000 Roskilde
Denmark
Telephone +45 46774004
bibl@risoe.dk
Fax +45 46774013
www.risoe.dk

Contents

Acknowledgements

I would like to thank Bent F. Srensen for readily available input and suggestions regarding cohesive zone models. A special thanks is going to Lars Pilgaard Mikkelsen for all the help and discussions about finite element analysis and ABAQUS.

1 Introduction

For glass fibre composites, the interfacial properties are controlled by the sizing, which is applied to the glass fibres during manufacture. For the same matrix system, a change of sizing results in changes of these properties, thereby influencing the mechanical properties such as strength and fracture toughness. The concept of strength is used for characterising crack initiation in composite design, while fracture toughness determines crack growth and damage development. In mode I crack growth in unidirectional fiber composites, fibre cross-over bridging occurs during cracking along the fiber direction. This failure mode plays an important role during delamination of fibre composites and splitting cracks around holes and notches. The fibre bridging zone must be modelled as a discrete mechanism on its own; failure is not just controlled by the cracking at the crack tip. The failure process can be described by a bridging law, which defines the relationship between the crack opening displacement and the local bridging tractions resulting from the bridging ligaments. Cohesive laws were measured experimentally in previous work. This report derives the necessary basics and equations to implement these laws into the commercial finite element code ABAQUS with a cohesive user element. Different numerical adjustments of the bridging law are discussed in detail. Crack aspects, such as crack opening shape and the influence of bridging law parameters, are studied based on the numerical results. It is furthermore of interest to identify the experimental measurements which show the highest sensitivity with respect to the bridging law shape.

2 Mode I bridging law measurement

The approach for the measurements of bridging laws is based on the application of the path independent J integral [1], and has been used recently to determine the bridging characteristics of unidirectional carbon fibre/ epoxy composites [2] and glass fibre composites [3]. A symmetric DCB specimen is loaded with pure bending moments M (Figure 1) under pure mode I. This specimen is one of the few practical specimen geometries, for which the global J integral (i.e. the integral evaluated around the external boundaries of the specimen) can be determined analytically [1]:

$$J = 12(1-\nu_{13}\nu_{31})\frac{M^2}{b^2H^3E_{11}} \tag{1}$$

E_{11} is the Young's modulus referring to the material directions, ν_{13} and ν_{31} are the major and minor Poisson's ratio, b is the width and H the beam height.

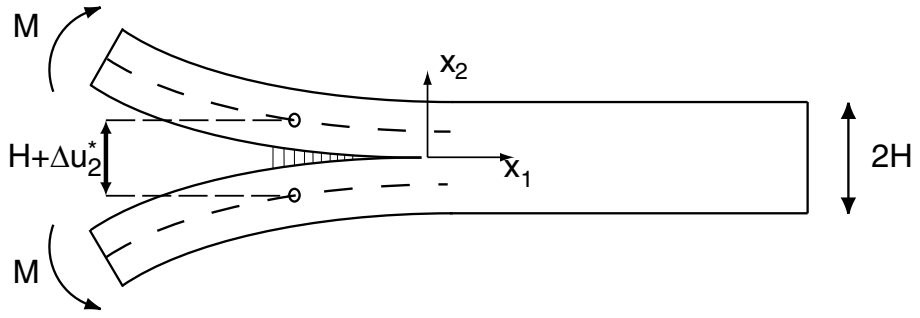


Figure 1. DCB specimen with pure bending moment

Now consider the specimen having a crack with bridging fibres across the crack faces near the tip. The closure stress σ (x_2-direction) can be assumed to depend only on the local crack

opening δ, i.e. the crack grows in pure mode I. The bridging law $\sigma = \sigma(\delta)$ is then taken as identical at each point along the bridging zone. Since fibres will fail when loaded sufficiently, we assume the existence of a characteristic crack opening δ_0, beyond which the closure traction vanishes. Shrinking the path of the J integral to the crack faces and around the crack tip [4] gives

$$J = \int_0^{\delta^*} \sigma(\delta) d\delta + J_{\text{tip}}, \tag{2}$$

where J_{tip} is the J integral evaluated around the crack tip (during cracking J_{tip} is equal to the fracture energy of the tip, J_0). The integral is the energy dissipation in the bridging zone and δ^* is the end-opening of the bridging zone at the notch root.

Connection is made to the overall R-curves as follows. By definition J_R is the value of J during crack growth. Initially, the crack is unbridged. Thus, by Eq. (2), crack growth initiates when $J_R = J_{\text{tip}} = J_0$. As the crack grows, J_R increases in accordance with Eq. (2). When the end opening of the bridging zone δ^* reaches δ_0, the overall R-curve attains its steady state value J_{ss}.

The bridging law can be determined by differentiating Eq. (2) [4].

$$\sigma(\delta^*) = \frac{\partial J_R}{\partial \delta} \tag{3}$$

The applied moment and the end opening of the bridging zone Δu_2^* are recorded. Assuming that $\delta^* \approx \Delta u_2^*$, where Δu_2^* is the notch opening measured at the neutral axis of the DCB specimen (see Figure 1), the bridging law can be determined. This approach models the bridging zone as a discrete mechanism on its own. Contrary to crack growth resistance curves (R-curves), the bridging law can be considered a material property and does not depend on specimen size [2].

The test above has been modified with two different bending moments to result in mixed mode testing [5]. In this case, mixed bridging laws can be measured. This has not been undertaken for the current material selection.

3 Experimental results

Recently, we have, by the use of a J integral based approach, measured the bridging laws under mode I fracture during transverse splitting of unidirectional glass-fiber/epoxy and glass-fiber/ polyester composites with different interface characteristics [3]. With increasing applied moment, crack propagation took place. Fibre cross-over bridging developed in the zone between the notch and the crack tip.

J_R is calculated according to Eq. (1). The specimen width b was 5 mm with a beam height of H=8 mm. Assuming that the unidirectional composite is transversely isotropic, the following elastic composite data were applied for Eq. (1) as previously measured: $E_{11,\text{epoxy}}$= 41.5 GPa, E_{33}= 9.2 GPa, $E_{11,\text{polyester}}$ = 42 GPa, $E_{33,\text{polyester}}$= 10 GPa and ν_{13}=0.3 (assumption). The analytical function

$$J_R(\delta^*) = J_0 + \Delta J_{\text{ss}} \left(\frac{\delta^*}{\delta_0} \right)^{\frac{1}{2}} \tag{4}$$

was found to fit all experimental data curves of crack growth resistance versus crack opening well, resulting in curve fits as shown in Figure 2. J_0 is the initial value of the experimental curve and equal to the fracture energy of the tip during crack growth, while ΔJ_{ss}, which is equal to $(J_{\text{ss}} - J_0)$, is the increase in crack growth resistance. Sørensen and Jacobsen [2] found that the same function fit the data of carbon fibre composite systems well.

The experimental values for the bridging laws are given in Table 1. The starting value J_0 indicates the point of crack growth initiation and can easily be determined during the experiment. The highest value of 345 J/m^2 was observed for the sizing B/ epoxy system. The crack

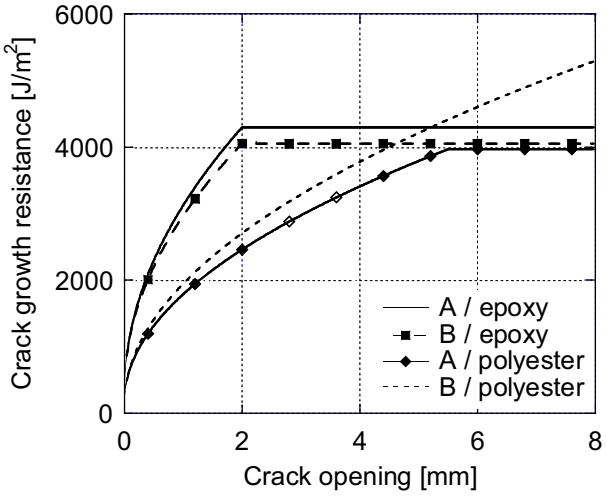


Figure 2. Comparison of crack growth resistance

Table 1. Experimental values for the bridging law for different composite systems

Composite system	J_0 [J/m^2]	ΔJ_{ss} [J/m^2]	δ_0 [mm]
sizing A/epoxy	321 ± 40	4000 ± 1000	2.0 ± 0.2
sizing B/epoxy	345 ± 30	3700 ± 500	2.0 ± 0.2
sizing A/polyester	150 ± 20	3800	5.5
sizing B/polyester	120 ± 30	> 4100	> 5.0

initiation value is significantly lower for the sizing B/ polyester system with 120 J/m^2, which is also related to a significantly lower transverse strength of not more than half the strength of the other composites [3]. In fact, the relation between transverse strength (measured with the transverse bending test [3]) and crack initiation for the different composite systems is fitted well by a linear relationship as seen in Figure 3. This verifies the assumption that the crack initiation value for the DCB test is controlled by the strength of the fiber-matrix bond. It can furthermore be seen that both axes show about the same ratio low strength and high strength of nearly a factor 3; however, the transverse bending test results in much lower standard deviations. The higher standard deviation for the DCB fracture initiation is most likely explained by specimen-to-specimen differences of the manufactured notch, which influences the crack initiation process.

The end opening value δ_0 at the onset of steady-state cracking was determined to be 2 mm for the epoxy systems. For the sizing B/ polyester system, steady-state cracking could not be determined with the present specimens, as the fibres continued to bridge the whole length of the crack after the maximum measurable notch opening of 5 mm was obtained. Since no upper bound was found for ΔJ_{ss}, this bridging behaviour was termed 'infinite toughening'.

Differentiating equation (4) results in the bridging law

$$\sigma(\delta) = \frac{\Delta J_{ss}}{2\delta_0} \left(\frac{\delta}{\delta_0}\right)^{-\frac{1}{2}}, \quad \text{for} \quad 0 < \delta < \delta_0, \tag{5}$$

where ΔJ_{ss} is the increase in crack growth resistance due to bridging (from zero to steady state bridging), and δ_0 is the crack opening where the bridging stress vanishes.

The bridging laws for the different fibre systems are compared in Figure 4. The bridging law can be considered a material property [2, 6] and is in an accessible form for implementation into finite element codes.

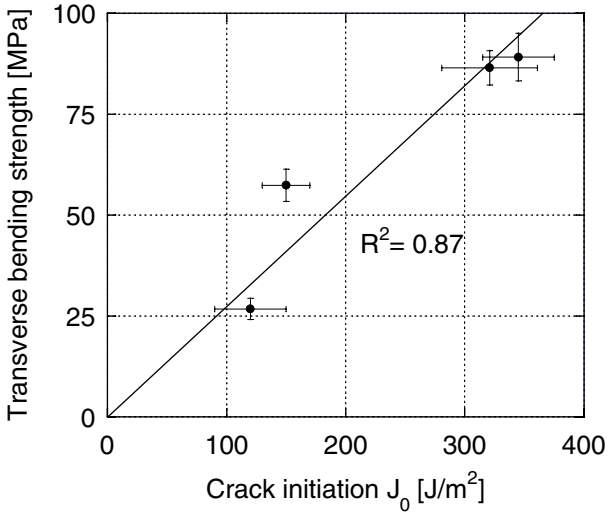


Figure 3. Relationship between crack initiation value J_0 and the transverse composite strength for the different composite systems

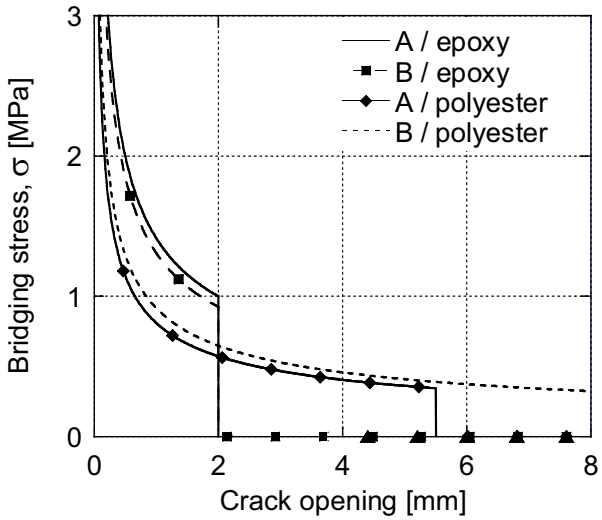


Figure 4. Comparison of resulting bridging laws

4 Implementation of the user element

There are a variety of possible methods for implementing cohesive laws within commercial finite element programs. The most versatile is the development and programming of cohesive elements [7–10]. These elements are in most cases defined with zero thickness and prescribe stresses based on the relative displacement of the nodes of the element. Similar work has also been undertaken with spring elements (force-opening relation), although in this case there might be simplifications required when calculating the equivalent nodal spring forces from the surrounding elements. The procedure is not straight forward when springs are connected to elements with non-linear shape functions, such as 8-noded elements [11].

4.1 Background and basic equations

Figure 5 shows the interface elements in 2-D and 3-D. The interface element is made up of two quadratic line elements (a), or two quadratic 2-D plane elements (b). These elements connect the faces of adjacent elements during the fracture process. The implementation is based on [10]. Quadratic elements are chosen as the cantilever beam mostly deforms under bending, which is best modelled by quadratic solid elements. The nodes of the interface element need to fit to these elements. The node numbering is chosen according to ABAQUS conventions of quadratic 2-D and 3D solids. The elements can also be derived in linear form by substituting the quadratic shape functions with linear ones in the appropriate equations. Consequently, the node numbers and degree of freedoms for the elements will decrease.

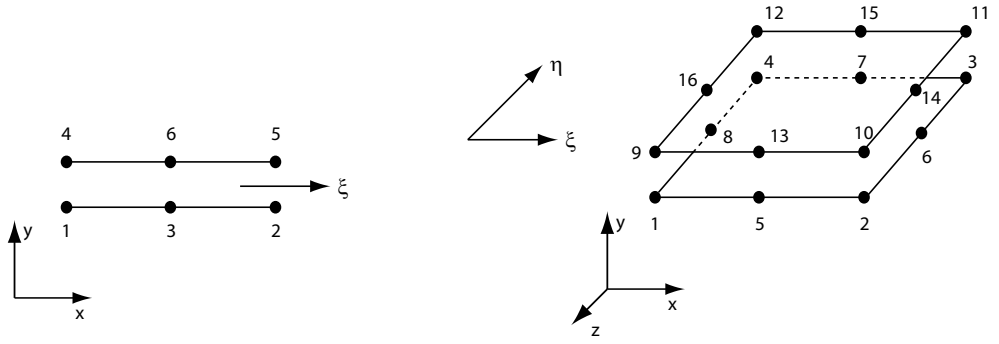


Figure 5. (a) Quadratic line interface element (3 node pairs) and (b) Quadratic plane interface element (8 node pairs)

The two surfaces of the interface element initially lie together in the unstressed deformation state (zero thickness) and separate as the adjacent elements deform. The relative displacements of the element faces create normal and shear displacements, which in turn generate element stresses depending on the constitutive equations (stress - opening relations) of the material. The constitutive relationship was derived in the experimental section, and is independent of the element formulation.

The implementation of a general interface element is explained in the following. Details are given for the special case of the quadratic line element for 2-D simulations, while the equivalent formulations for the quadratic plane element for 3-D simulations are provided in Appendix A.

The line interface element has 12 (2x6) degrees of freedom. The vector (12×1) of the nodal displacements in the global coordinate system is given as:

$$\mathbf{d}_N = \left(d_x^1 \;\; d_y^1 \;\; d_x^2 \;\; d_y^2 \;\; \cdots \;\; d_x^6 \;\; d_y^6\right)^{\mathrm{T}} \tag{6}$$

The order follows typical ABAQUS' conventions, and this is considered in the derived formulations below.

The opening of the interface element is defined as the difference in displacements between the top and bottom nodes:

$$\Delta u = \{u\}^{\text{top}} - \{u\}^{\text{bot}}, \tag{7}$$

thereby leading to the following definition of the interface opening $\Delta\mathbf{u}_N$ in terms of nodal displacements of paired nodes:

$$\Delta\mathbf{u}_N = \mathbf{\Phi}\mathbf{d}_N = \left[-\mathbf{I}_{6\times6} \quad | \quad \mathbf{I}_{6\times6}\right]\mathbf{d}_N \tag{8}$$

where $\mathbf{I}_{6\times6}$ denotes a unity matrix with 6 rows and columns. $\mathbf{u}_N$ is a 6×1 vector.

From the nodal positions, the crack opening is interpolated to the integration points with the help of standard shape functions. Let $N_i(\xi)$ be the shape functions for the node pair i $(i = 1, 2, 3)$, where ξ stands for the local coordinate of the element with $-1 \leq \xi \leq 1$. The

relative displacement between the nodes for each point within the element is then given by:

$$\Delta\mathbf{u}(\xi) = \begin{pmatrix} \Delta u_x(\xi) \\ \Delta u_y(\xi) \end{pmatrix} = \mathbf{H}(\xi)\Delta\mathbf{u}_N, \tag{9}$$

where $\mathbf{H}(\xi)$ is a 2x6 matrix containing the quadratic shape functions.

For the line element, it is of the form

$$\mathbf{H}(\xi) = \begin{pmatrix} N_1(\xi) & 0 & N_2(\xi) & 0 & N_3(\xi) & 0 \\ 0 & N_1(\xi) & 0 & N_2(\xi) & 0 & N_3(\xi) \end{pmatrix}. \tag{10}$$

The shape functions for the line element are given in Appendix B. As a result, we get

$$\Delta\mathbf{u}(\xi) = \mathbf{H}(\xi)\mathbf{\Phi}\mathbf{d}_N = \mathbf{B}(\xi)\mathbf{d}_N, \tag{11}$$

where $\mathbf{B}(\xi)$ is of the dimension 2x12 and $\Delta\mathbf{u}(\xi)$ of the dimension 2x1; thereby describing the continuous displacement field in both directions within the element.

During large deformations, the element requires a local coordinate system to compute local deformations in normal and tangential directions. The most common choice is a coordinate system given by the middle points of the two element faces, which thereby coincides with the nodal positions in the undeformed state. If the coordinates of the initial configuration are given by the vector $\mathbf{x}_N$ and the deformation state is defined by the vector $\mathbf{d}_N$, the reference surface coordinates $\mathbf{x}_N^R$ are computed by linear interpolation between the top and bottom nodes in their deformed state:

$$\mathbf{x}_R^N = \frac{1}{2}(\mathbf{I}_{6\times6} \quad | \quad \mathbf{I}_{6\times6})(\mathbf{x}_N + \mathbf{d}_N) \tag{12}$$

The coordinates of any specific reference plane can be derived analogous to Eq. (9):

$$\mathbf{x}^R(\xi) = \begin{pmatrix} x^R(\xi) \\ y^R(\xi) \end{pmatrix} = \mathbf{H}(\xi)\mathbf{x}_N^R \tag{13}$$

This local coordinate vector, with unit length, is obtained by differentiating the global position vector with respect to the local coordinates:

$$\mathbf{t}_1 \quad = \quad \frac{1}{||\frac{\partial\mathbf{x}^R}{\partial\xi}||}\left(\frac{\partial x^R}{\partial\xi}, \frac{\partial y^R}{\partial\xi}\right)^{\mathrm{T}}. \tag{14}$$

The normal vector (also with unit length) of the local coordinate element needs to be perpendicular to the vector $\mathbf{t}_1$:

$$\mathbf{t}_{\mathrm{n}} = \frac{1}{||\frac{\partial\mathbf{x}^R}{\partial\xi}||}\left(\frac{-\partial y^R}{\partial\xi}, \frac{\partial x^R}{\partial\xi}\right)^{\mathrm{T}}, \tag{15}$$

and the derivatives are determined as follows:

$$\frac{\delta\mathbf{x}^R(\xi)}{\delta\xi} = \begin{pmatrix} x^R_{,\xi} \\ y^R_{,\xi} \end{pmatrix} = \frac{\delta\left(\mathbf{H}(\xi)\right)}{\delta\xi}\mathbf{x}_N^R = \mathbf{h}(\xi)\mathbf{x}_N^R, \tag{16}$$

with

$$\mathbf{h}(\xi) = \begin{pmatrix} N_{1,\xi}(\xi) & 0 & N_{2,\xi}(\xi) & 0 & N_{3,\xi}(\xi) & 0 \\ 0 & N_{1,\xi}(\xi) & 0 & N_{2,\xi}(\xi) & 0 & N_{3,\xi}(\xi) \end{pmatrix}. \tag{17}$$

The derivatives of the shape functions are given in Appendix B.

The length of the vector is given by the standard definition:

$$||\frac{\partial\mathbf{x}^R}{\partial\xi}|| = \sqrt{\left(\frac{\partial x^R}{\partial\xi}\right)^2 + \left(\frac{\partial y^R}{\partial\xi}\right)^2}. \tag{18}$$

The components $\mathbf{t}_1$ and $\mathbf{t}_n$ represent the direction cosines of the local coordinate system to the global one, thus defining the 2×2 transformation tensor $\mathbf{\Theta}$:

$$\mathbf{\Theta} = [\mathbf{t}_1, \mathbf{t}_n], \tag{19}$$

which relates the local and global displacements as follows:

$$\Delta \mathbf{u}_{\text{loc}} = \mathbf{\Theta}^{\text{T}} \Delta \mathbf{u}. \tag{20}$$

In the following, local matrices will be indicated by the subscript $_{\text{loc}}$ as above, while the equations otherwise refer to the global values.

$\mathbf{t}_{\text{loc}}$ is the 1×2 vector defining the bridging stresses in the local coordinate system and relates to the local relative displacement via the constitutive expression for the interface element:

$$\mathbf{t}_{\text{loc}} = \begin{pmatrix} \sigma_1 \\ \sigma_n \end{pmatrix} = \mathbf{C}_{\text{loc}}(\mathbf{\Delta u}_{\text{loc}})\ \mathbf{\Delta u}_{\text{loc}} \tag{21}$$

The constitutive expression can be expressed either with a linear displacement term for $\mathbf{\Delta u}$ as shown above, or with a coupled form, where $\mathbf{\Delta u}$ is included with non-linear dependence. The preferred option depends on the form of the constitutive equation. For our expression as introduced in Eq. (5), a coupled form is preferable:

$$\mathbf{t}_{\text{loc}} = \mathbf{C}_{\text{loc}}\ \mathbf{\Delta u}_{\text{loc}}^{-\frac{1}{2}} \tag{22}$$

$\mathbf{C}_{\text{loc}}$ is now a constant, and does not depend on the displacement. Note that in comparison with Eq. (5), δ has been replaced with the general numerical nomenclature for the opening, $\mathbf{\Delta u}_{\text{loc}}$. This convention will be kept in the following.

The element stiffness matrix and the right hand side nodal force vector are required for the UEL subroutine in ABAQUS.

The element force vector is of size 12×1. Its contribution to the global force vector is defined as

$$\begin{aligned} \mathbf{f}_N^{\text{el}} &= \int_{\text{A}_{\text{el}}} \mathbf{B}^{\text{T}} \mathbf{t}\, dA && (23)\\ &= W \int_{\text{L}_{\text{el}}} \mathbf{B}^{\text{T}} \mathbf{t}\, dl && (24)\\ &= W \int_{-1}^{1} \mathbf{B}^{\text{T}} \mathbf{t}\, \text{det}\mathbf{J}\, d\xi && (25)\\ &= W \int_{-1}^{1} \mathbf{B}^{\text{T}} \mathbf{\Theta} \mathbf{t}_{\text{loc}}\, \text{det}\mathbf{J}\, d\xi, && (26) \end{aligned}$$

where W is the width of the interface element and, as in most cases of 2-D modelling, also the width of the finite element model.

$\text{det}\mathbf{J}$ is the Jacobian defined by the transformation of the global coordinates (x, y) to the current element coordinate (ξ), and results, for the line element, in the same expression as previously used for calculating the length of the unit vector in Eq. (18):

$$\text{det}\mathbf{J} = \sqrt{\left(\frac{\partial x^R}{\partial \xi}\right)^2 + \left(\frac{\partial y^R}{\partial \xi}\right)^2}. \tag{27}$$

It should be noted that the Jacobian will in most analysis cases not be constant, but depend on the local element coordinates. It therefore needs to be derived for each integration point.

The tangent stiffness matrix of according size 12×12 (note negative sign convention for ABAQUS) is defined as

$$\mathbf{K}^{\text{el}} = -\frac{\partial \mathbf{f}_N^{\text{el}}}{\partial \mathbf{d}^{\text{el}}} \tag{28}$$

With the derivation from Eq. (23), this results in:

$$\mathbf{K} = -\int_{\mathrm{A_{el}}} \mathbf{B}^{\mathrm{T}} \boldsymbol{\Theta} \frac{\partial \mathbf{t}_{\mathrm{loc}}}{\partial \mathbf{d}^{\mathrm{el}}} \, dA \tag{29}$$

$$= -W \int_{\mathrm{L_{el}}} \mathbf{B}^{\mathrm{T}} \boldsymbol{\Theta} \frac{\partial \mathbf{t}_{\mathrm{loc}}}{\partial \mathbf{d}^{\mathrm{el}}} \, dl \tag{30}$$

$$= -W \int_{\mathrm{L_{el}}} \mathbf{B}^{\mathrm{T}} \boldsymbol{\Theta} \frac{\partial \mathbf{t}_{\mathrm{loc}}}{\partial \boldsymbol{\Delta} \mathbf{u}_{\mathrm{loc}}} \frac{\partial \boldsymbol{\Delta} \mathbf{u}_{\mathrm{loc}}}{\partial \boldsymbol{\Delta} \mathbf{u}} \frac{\partial \boldsymbol{\Delta} \mathbf{u}}{\partial \mathbf{d}^{\mathrm{el}}} \, dl \tag{31}$$

$$= -W \int_{\mathrm{L_{el}}} \mathbf{B}^{\mathrm{T}} \boldsymbol{\Theta} \frac{\partial \mathbf{t}_{\mathrm{loc}}}{\partial \boldsymbol{\Delta} \mathbf{u}_{\mathrm{loc}}} \boldsymbol{\Theta}^{\mathrm{T}} \mathbf{B} \, dl \tag{32}$$

$$= -W \int_{-1}^{1} \mathbf{B}^{\mathrm{T}} \boldsymbol{\Theta} \mathbf{D}_{\mathrm{loc}} \boldsymbol{\Theta}^{\mathrm{T}} \mathbf{B} \, \mathrm{det} \mathbf{J} \, d\xi \tag{33}$$

As can be seen above, the stiffness matrix $\mathbf{D}$ is defined as:

$$\mathbf{D}_{\mathrm{loc}} = \frac{\partial \mathbf{t}_{\mathrm{loc}}}{\partial \boldsymbol{\Delta} \mathbf{u}_{\mathrm{loc}}}. \tag{34}$$

It can also be expressed in terms of the constitutive expression in Eq. (21) with a linear dependence on $\boldsymbol{\Delta}\mathbf{u}$:

$$\mathbf{D}_{\mathrm{loc}} = \frac{(\partial \mathbf{C}(\boldsymbol{\Delta}\mathbf{u})\boldsymbol{\Delta}\mathbf{u})}{\partial \boldsymbol{\Delta}\mathbf{u}} = \frac{\partial \mathbf{C}(\boldsymbol{\Delta}\mathbf{u})}{\partial \boldsymbol{\Delta}\mathbf{u}} \boldsymbol{\Delta}\mathbf{u} + \mathbf{C}(\boldsymbol{\Delta}\mathbf{u}) \tag{35}$$

The local traction matrix $\mathbf{D}$ is then given by

$$\mathbf{D}_{\mathrm{loc}} = \begin{pmatrix} D_1 & D_c \\ D_c & D_n \end{pmatrix}. \tag{36}$$

The terms D_1 and D_n are derived by finding the derivatives according to Eq. (34). The components D_c are possible coupling terms. They are normally obtained if the traction laws are derived from an overall elastic potential for the cohesive law. For mode I loading, the relative displacement in u_1 direction will be zero, and thereby will lead to zero traction stresses. The coupling terms can then be set to zero as they do not influence the results. However, a dummy value for D_1 should be assigned to avoid possible numerical problems due to a singular stiffness matrix.

4.2 Unloading

The cohesive relationship as given in Eq. (21) is elastic, and the stresses transferred through the crack obey the same law whether the crack opens or closes. For most damage mechanisms, this is most likely not true, as it neglects the damage introduced at the interface during partial opening. Local unloading can occur when sudden changes occur in the external loading application or internal stress redistribution in neighbouring elements is introduced, for example by sudden failure. This problem is usually circumvented by introducing a maximum damage parameter Δu^*, which stands for the maximum value the opening Δu obtained during a given increment of the analysis. If the next value of Δu is larger, damage continues to grow; otherwise elastic unloading is assumed to occur (see the elastic unloading towards the origin in the simplified bridging law in Figure 6). This elastic mechanism applies for the case of fibre bridging. The failure mechanism includes peeling off of the fibers on either side of the crack, the development of bridging fiber ligaments and failure of these ligaments. During unloading, however, these ligaments simply deform elastically, thereby leading to a elastic unloading towards the origin, which can be observed during the experiment. For numerical purposes it should be noted that Δu^* should only be updated at the end of the increment; the current iterative opening solution is not considered.

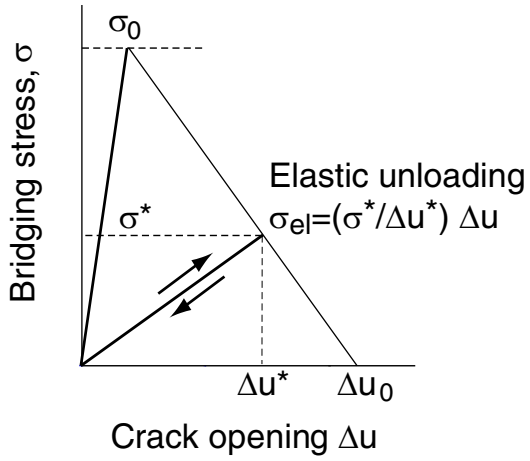


Figure 6. Elastic unloading after damage onset

4.3 State variables for the analysis

The number of state variables for each element depends on the chosen integration procedure. At each integration point of the linear element, there are two state variables for the relative displacement (x- and y-direction) and the local stresses acting in the x- and y-direction. Furthermore, there is one value describing the current interaction state between the two surfaces for the purpose of stability analysis and position tracking during the numerical procedure. For a standard integration procedure with 3 integration points, this results in 15 (3x5) state variables for the linear element. As ABAQUS will not terminate the analysis if an insufficient number of state variables is provided for the chosen integration procedure, this user input error is checked within the UEL subroutine.

4.4 Numerical integration

The integration scheme was previously shown to have an influence on the performance of the interface element. For the Newton-Cotes scheme, the integration points are located at the nodes. For linear elements, the application of the standard Gauss integration was shown to result in a coupling between the degrees of freedom of different node sets and then in oscillations of the traction profile in the presence of large traction gradients, such as during the initial stiffness increase prior to damage, over the element [7].

Non-convergence can occur if the element size is too large compared to the stress uptake based on the traction law: around three elements should be present to resolve the changing stress state in the interface [12]. This restriction on maximum element size can also be circumvented by choosing a larger number of integration points instead of reducing the element size [13]. These approaches are investigated in the report. The 3 point rules (normally chosen as standard in accordance with the quadratic element type) as well as 6 and 12 point rules for Gauss and Newton-Cotes were tested with the current model. The respective points and weights are given in Appendix C.

5 Numerical model

The beam-contact problem is shown in Figure 7. Owing to symmetry, it is sufficient to consider

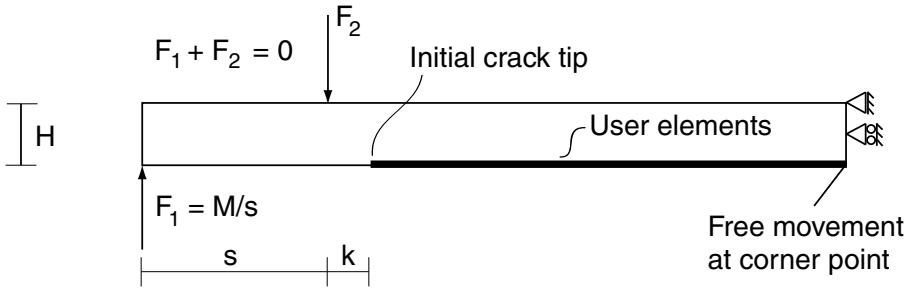


Figure 7. Problem statement and boundary conditions for pure moment loaded DCB specimen (symmetric half model)

only one beam of the DCB specimen, although full models were run to check the boundary conditions (see Section 5.1). The thickness H is equal to 8 mm, as discussed in Section 3. Plane strain conditions were assumed, which neglect edge effects. The mesh consisted of eight-noded plane strain solid CPE8 elements, which are suited to describe bending deformations without hourglass effects. The composite material is assumed to be transversely isotropic. The material properties were given in the previous section. Nonlinear, large displacements are considered in the analysis to rotate the non-isotropic material properties accordingly. The interface elements are applied from the crack tip onwards to the end of the beam. The lower element nodes are fixed on the symmetry line.

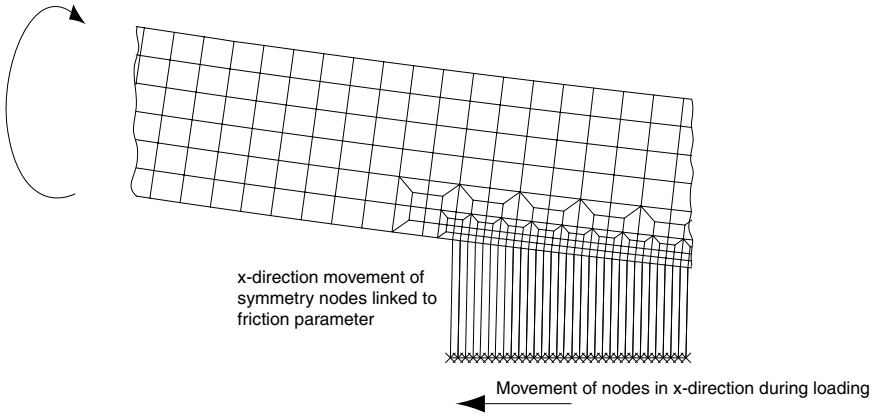
5.1 Symmetric half model versus full model: boundary conditions

For the symmetric half model, additional nodes are duplicated on the symmetry line for the description of the user elements. During deformation, the top and bottom node sets will separate. The bottom nodes are therefore fixed in the y-direction as a zero boundary condition. However, preliminary analyses showed that this condition does not truly represent the full model (see Figure 8), as the nodes can move freely along the symmetry line in the x-direction. For the true full model, the bottom nodes will move in the same way as the top nodes, therefore maintaining the same absolute x-value (no shear introduced in the elements). As discussed previously, the coupling terms in the local constitutive equation (see Eq. (36)) therefore do not influence the results. For the symmetric half model, the movement of the symmetry nodes in x-direction therefore needs to be coupled to the movement of the beam nodes via *EQUATION for each duplicated node. The difference in deformation is visualised in Figure 8. The effect is rather small, but leads to differences in the results for larger beam deflection.

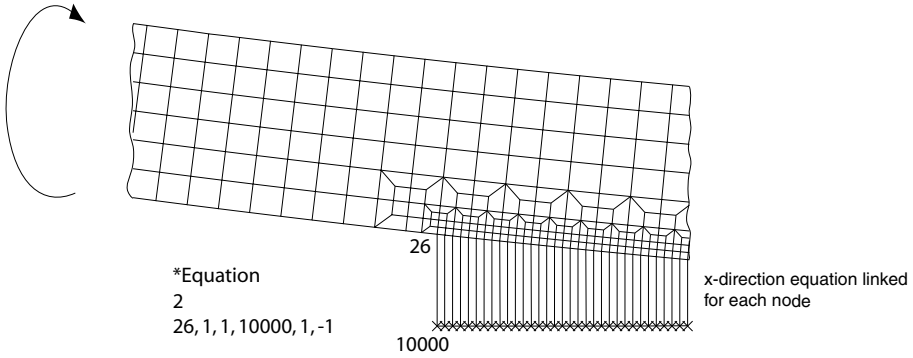
5.2 Application of pure moment bending by displacement control

In analyses with possible decreasing stresses due to introduced material damage or decreasing bridging stresses, displacement controlled deformation, rather than force controlled, becomes the preferred way of introducing boundary conditions. Furthermore, displacement control is mostly applied in experimental testing and also in the case of the DCB specimen testing: testing procedure and simulation are therefore more closely related.

The application of a homogeneous bending moment by force control requires consideration



(a) Free movement of symmetry nodes



(b) Linked movement of symmetry nodes

Figure 8. Effekt of linking symmetry nodes

of the element type, where varying forces need to be applied to each node according to the underlying shape functions of the element type. This procedure does unfortunately not work for displacements, as for a given translation the rotation of the beam is unknown. The displacement controlled procedure introduced in the following was first described in previous work of the group [11]. The moment application is simplified, thereby resulting in a non-homogeneous displacement field towards the nodes of the applied boundary conditions at the end of the beam. However, pure bending conditions are introduced around the contact zone by ensuring that the distance k between moment introduction and beginning of the contact zone (see Figure 7) is large enough. The value is set to k=12 mm in the present model, with s being equal to 24.65 mm. As a rule of thumb, boundary conditions should be applied more than the beam thickness away from the point of interest to ensure a uniform bending stress in this region. With a beam height of H=8 mm, this requirement is fulfilled.

The displacements v_1 and v_2 of two nodes are to be controlled such that the resulting forces in the two nodes, F_1 and F_2, are equal and opposite in magnitude, thereby introducing a pure bending moment (see Figure 9). To accomplish this, v_1 and v_2 are inter-connected by two so-called dummy nodes A and B, having displacement v_A and v_B, respectively. Dummy nodes are nodes that are not associated with the geometry. Their positions with respect to the beam structure are arbitrary, but their displacements have to be coupled to the structural deformation. In the following we derive a suitable relationship between v_1, v_2, v_A and v_B. The displacement of node A is defined as

$$v_A = v_1 - v_2. \tag{37}$$

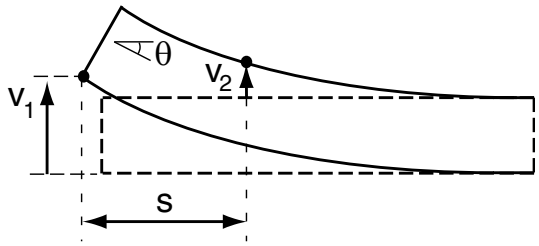


Figure 9. Displacements used for controlling the rotation and applying a pure moment to the beam arm of a DCB specimen

Note that $\theta = \arctan(v_A/s) \approx v_A/s$ is the rotation of the beam, with s being the current difference in x-coordinates between nodes 1 and 2, i.e. the moment arm. The distance between the nodes will change during the analysis, which needs to be taken into account. The displacement in node B is set to

$$v_B = v_1 + v_2. \tag{38}$$

These definitions are mostly arbitrary. In a physical sense, they can be described as a rotational component (displacement ν_A) and a translational component (displacement ν_B). We have to make sure that the same energy is applied to the dummy nodes as is applied to the structure itself. Using the principle of virtual work, the applied incremental elastic energy dw is given by

$$dw = F_1 dv_1 + F_2 dv_2. \tag{39}$$

Similarly, the incremental elastic energy applied to nodes A and B is

$$dw = F_A dv_A + F_B dv_B, \tag{40}$$

where F_A and F_B are the forces in nodes A and B. Finally, the pure bending condition

$$F_1 + F_2 = 0 \tag{41}$$

must hold true. As the forces in the dummy nodes must (according to the principle of virtual work) perform the same incremental work as F_1 and F_2, the following relations are obtained:

$$dw = F_A d(v_1 - v_2) + F_B d(v_1 + v_2) \tag{42}$$

$$= (F_A + F_B)dv_1 + (F_B - F_A)dv_2 \tag{43}$$

In comparison with Eq. (39), this gives the correct relationships between the forces at the nodes.

$$F_1 = F_A + F_B \quad \text{and} \quad F_2 = F_B - F_A \tag{44}$$

$$2F_A = F_1 - F_2 \quad \text{and} \quad 2F_B = F_1 + F_2 \tag{45}$$

Comparing Eq. (41) and Eq. (44) leads to $F_B = 0$, and gives $F_A = F_1$. Thus, the constraints between v_1, v_2, v_A and v_B in Eq. (37) and Eq. (38) provide the correct constraints. The multi-point constraints for the numerical analysis are implemented into the finite element analysis as follows:

$$2v_1 - v_A - v_B = 0 \quad \text{and} \quad 2v_2 + v_A - v_B = 0 \tag{46}$$

In summary, F_A is the applied force to the beam from which the moment $M = F_A s = F_1 s$ can be calculated for each increment. J_R is computed from Eq. (1) using the calculated moment. The displacement v_A is the loading parameter, which is increased during the simulations.

5.3 Bridging law adjustments for numerical modelling

The bridging law adjustment is shown exemplary for the case of the normal stresses across the interface element. With the experimental result of Eq. (5), the stress σ_n in normal direction was defined by the following non-linear relationship

$$\sigma_n = K_n \, \Delta u_n^{-\frac{1}{2}} \quad \text{with} \quad C_n = \frac{\Delta J_{\text{ss}}}{2\sqrt{\Delta u_{0,n}}} \quad \text{for} \quad 0 < \Delta u_n < \Delta u_{0,n}. \tag{47}$$

C_n is a constant in this case as explained in Eq. (22). In the following, we will drop the subscript n for the normal direction. The relationship above can easily be modified or be expressed in piecewise linear form for different values of Δu. The use of numerical predictions for a better fit of the cohesive law to the experimental results will be discussed later.

Two points need to be addressed during the numerical adjustment:

- Removal of the stress singularity at $\Delta u = 0$ and
- Incorporation of the intial fracture strength J_0.

The bridging law as shown in Eq. (47) has a simple form, but inhibits a singularity in $\Delta u = 0 : \sigma \rightarrow \infty$ (see Eq. 5). Two different methods of numerical adjustment are thought of in the following as shown in Figure 10. Micromechanical considerations [6] predict that crack initiation starts at a finite stress value at the interface, which is reached as the deformation starts to take place. At the point of crack initiation, the opening at the interface will still be zero (a). This option is unconventional in the numerical sense; and the implications of using a non-zero starting value for the analysis results have been explored previously [14]. In the case of a finite stress value σ_0 for $\Delta u = 0$ in the bridging law, the finite element model actually starts with a force imbalance, as no initial force equilibrium is given. This is resolved in the first increment by overlapping the user elements to achieve zero stresses away from the crack tip. From a numerical point of force equilibrium at analysis start, a zero stress state is preferable at $\Delta u = 0$ as shown in (b). For small values of Δu_1 and $Deltau_2$ these adjustments are only have a small influence on the numerical results.

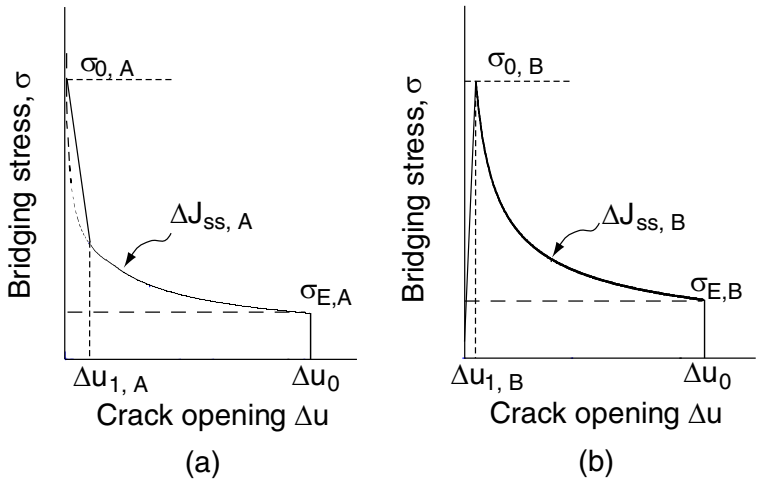


Figure 10. Possibilities of circumventing infinite stress at zero opening

For both methods, the most important point is that the energy uptake is equivalent in the experimental and numerical model. Details of the bridging law can then be adjusted in the numerical evaluation to provide the best fit with the experimental data.

The differentiation of Eq. (4), which leads to Eq. (5), results in a loss of the initial starting value J_0 (see Table 1). It is therefore not advisable to simply use the earlier bridging law

parameters for the numerical simulations. A further adjustment is required. Two procedures based on the initial choice of singularity treatment above are proposed.

The total energy uptake of the experimental bridging law is determined by calculating the area W_{exp} under the curve.

$$\sigma_{\text{exp}} = \frac{\Delta J_{\text{ss}}}{2\Delta u_0}\left(\frac{\Delta u_0}{\Delta u}\right)^{\frac{1}{2}} \tag{48}$$

$$W_{\text{exp}} = \Delta J_{\text{ss}} + J_0 \quad \text{for} \quad \Delta u \in [0, \Delta u_0] \tag{49}$$

For a better physical understanding, J_0 can be considered as the elastic fracture energy during initial fracture (strength controlled), while ΔJ_{ss} represents the damage fracture energy (toughness controlled). To ensure the same energy uptake during the simulated fracture process, the initial starting value J_0 needs to be incorporated into the bridging law.

For the two bridging law adjustments introduced in Figure 10, different ideas for including J_0 are suggested. For method (a) with a finite stress, the initial linear bridging law part can be further adjusted in such a way that the numerical fracture energy at the point Δu_1 of the bridging process also includes the experimental starting value J_0, while for method (b), the easiest method is to split the two components J_0 and ΔJ_{ss} within the bridging law, and use the initial increase up to Δu_2 to introduce J_0. In the following, the necessary adjustments are derived for both cases, and the results are compared with the experimental data.

Finite stress value

Figure 11 visualises method (a). Upon reaching the value of Δu_1, the energy uptake is adjusted

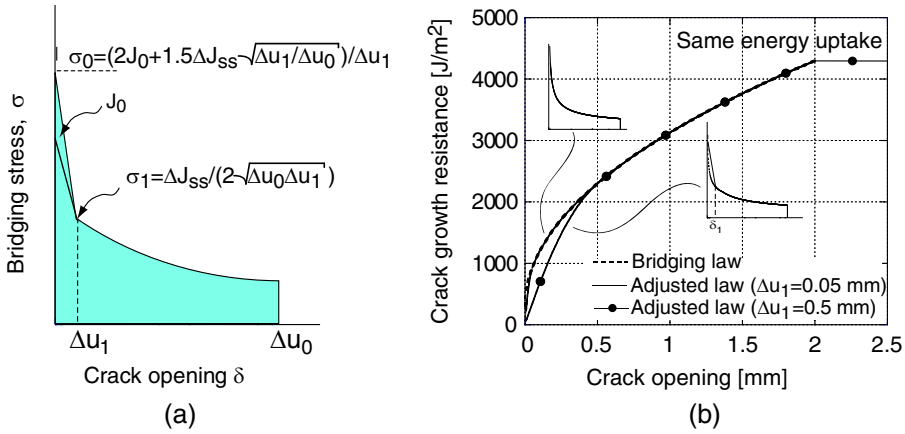


Figure 11. Numerical bridging law (adjusted). Combined total fracture energy.

according to the experimental curve:

$$W_{\text{exp}} = \Delta J_{\text{ss}}\left(\frac{\Delta u_1}{\Delta u_0}\right)^{\frac{1}{2}} + J_0 \quad \text{for} \quad \Delta u \in [0, \Delta u_1] \tag{50}$$

The maximum stress σ_0 is found by requiring that the area W_1 of the linear softening law in the range $0 \leq \Delta u \leq \Delta u_1$ must equal the area W_{exp} of the general bridging law in this range (see Eq. (50)). Area W_1 is given by

$$W_1 = \frac{\Delta J_{\text{ss}}}{2}\left(\frac{\Delta u_1}{\Delta u_0}\right)^{\frac{1}{2}} + \frac{1}{2}\sigma_0 \Delta u_1 \tag{51}$$

Setting $W_{\text{exp}} = W_1$, while including the starting value J_0 as before, the maximum stress for a given Δu_1 (as also included in Figure 11) is

$$\sigma_0 = \frac{2J_0 + \frac{3}{2}\Delta J_{\text{ss}} \left(\frac{\Delta u_1}{\delta_0}\right)^{\frac{1}{2}}}{\Delta u_1} \tag{52}$$

The energy uptake J_0 is now included by raising the starting value σ_0. Upon reaching of Δu_1, the original bridging law is followed.

Zero stress value

In the second case of a zero stress condition with subsequent increase of the bridging stresses, a separated bridging law is proposed. A small value of Δu_2 is assumed, up to which the energy J_0 is taken up. This is the physically more relevant case, and is visualised in Figure 12.

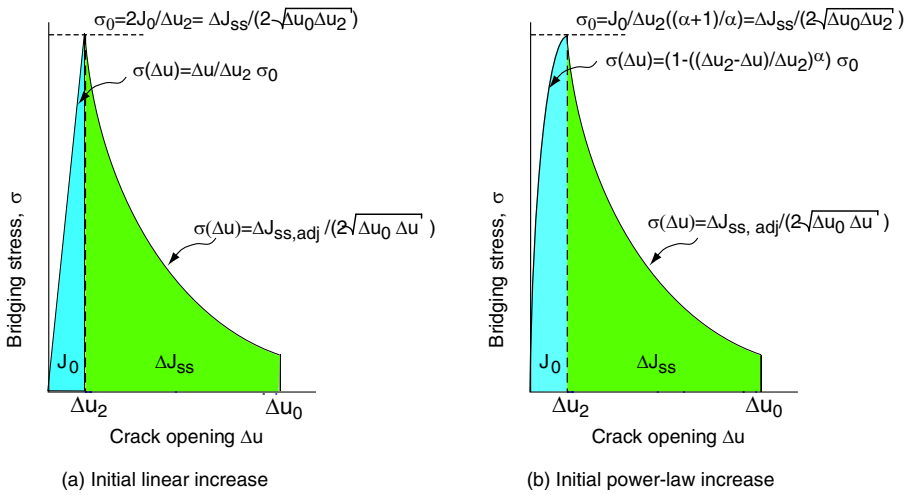


Figure 12. Numerical bridging law (adjusted). Separated fracture energy.

The value of Δu_2 is determined from the initial fracture energy of J_0 and the maximum stress value, σ_0. Δu_2 can become quite small based on the value of J_0. This leads to convergence problems with the linear increase shown in Figure 12(a), as a steep tangent changes to a negative tangent at $\Delta u = \Delta u_2$. Instead of the common linear increase up to Δu_2, a power-law approach (see Figure 12(b)) can be applied:

$$\sigma(\Delta u) \quad = \quad \sigma_0 \left(1 - \left[\frac{\Delta u_2 - \Delta u}{\Delta u_2}\right]^{\alpha}\right) \tag{53}$$

Δu_2 can then be determined for a given value of J_0 by the integral:

$$\int_0^{\Delta u_2} \sigma(\Delta u) d\Delta u = J_0 \tag{54}$$

$$\Rightarrow \Delta u_2 = \frac{J_0}{\sigma_0}\left(\frac{\alpha+1}{\alpha}\right) \tag{55}$$

The power-law factor α determines the initial increase in stress with opening. The increase with $\alpha = 1$ is identical to the linear increase. A factor of $\alpha = 100$ leads to the same result for Δu_2 as a constant finite stress value up to Δu_2 with a non-equilibrium starting point as shown in Table 2. The proposed power-law function is therefore more versatile. However, the power-law approach is also a numerically more robust approach as the sign change in the tangent stiffness

Table 2. Comparison of values for Δu_2 for a maximum stress of $\sigma_0 = 3.5$ MPa and $J_0 = 300$ J/m^2

	α=1	α=5	α=10	α=100	linear	const σ_0
Δu_2 [mm]	0.1714	0.1029	0.0943	0.0857	0.1714	0.0857

at $\Delta u = \Delta u_2$ is avoided. The tangent is close to zero before becoming negative. This leads to largely improved numerical behaviour in terms of convergence.

As the energy uptake for the damage part needs to equal the original ΔJ_{ss} for whole bridging law between 0 and Δu_0, the value of $\Delta J_{\text{ss, adj}}$ (see Figure 12) in the bridging law needs to be adjusted accordingly:

$$\Delta J_{\text{ss, adj}} = \frac{\Delta J_{ss}}{1 - \sqrt{\frac{\Delta u_2}{\Delta u_0}}} \tag{56}$$

The problem with this procedure is that σ_0 in turn again depends on the adjusted value for $\Delta J_{\text{ss, adj}}$:

$$\sigma_0 = \frac{\Delta J_{\text{ss, adj}}}{2\sqrt{\Delta u_2 \Delta u_0}} \tag{57}$$

As a consequence, σ_0 and therefore Δu_2 will change as well. An iterative procedure needs to be applied to determine the values for $\Delta J_{\text{ss, adj}}$ and Δu_2. A starting value for Δu_2 in Eq. (56) can be determined by integrating the bridging law within the limits of Δu_2 and Δu_0 and setting the integral value equal to ΔJ_{ss} (see Figure 12 (b)):

$$\Delta u_2 = \frac{4 J_0^2 \Delta u_0}{\Delta J_{ss}^2} \left(\frac{\alpha + 1}{\alpha} \right)^2 . \tag{58}$$

We therefore determine first Δu_2 with Eq. (58). This is followed by calculating $\Delta J_{\text{ss, adj}}$ with Eq. (56) and σ_0 is calculated with Eq. (57). In the last step, Δu_2 is determined with Eq. (55). The latter three steps are repeated until convergence is achieved.

6 Numerical results

In the following sections, results are introduced for the bridging law shape with finite stress value (method (a)) and the bridging law shape with split values of J_0 and ΔJ_{ss} with a power-law increase (method (b)) as introduced earlier. Based on these results, the bridging law shape was then modified to give a better fit with the experimental data. To justify this procedure, it is important to keep in mind that the bridging law was derived in the first place by analytical curve fitting of the crack growth resistance (see Section 2). Any deviation from the experimental data to this curve fitting will of course influence the match of the predictions. Analytically, it is only possible to fit the crack growth resistance. In this section, we also compare the numerical fit with other experimental observations, such as the crack growth resistance as a function of crack growth development and the shape of the crack opening. Based on these findings, it is shown that the crack growth resistance as a function of crack growth development is much more sensitive to changes in the bridging law shape. Numerical adjustment of the bridging law parameters on this curve leads to a very good agreement of the numerical predictions with all measured experimental data.

6.1 Comparison of numerical integration procedures

Non-convergence problems can occur in the following simulations if the element size is too large compared to the stress uptake based on the traction law. This is most severe for the method of the finite stress value, while all approaches with the power-law show a better numerical

performance. This restriction on element size can be improved by choosing a larger number of integration points instead of reducing the element size.

Different mesh refinements were used to determine the numerical response during the simulations. For the first refinement, which was shown in Figure 8, the element length for the cohesive elements is given by 0.36 mm. This is in relation to values of $\Delta u_1 = 0.05\ mm$ and $\Delta u_2 \approx 0.01\ mm$. The mesh density is not sufficient to ensure convergence for 3 Gauss integration points and Newton-Cotes points for all methods introduced in the following: further mesh refinement for those integration procedures is required down to $l_{el} = 0.0906\ mm$ is required. At this level, all integration procedures converge.

However, the use of 6 integration points for both the Gauss and Newton-Cotes integration result in convergence of the simulation. Running the simulation with a lower mesh density is of course beneficial in terms of computational effort. For 12 integration points, Gauss integration again shows good performance, while the 12 point Newton-Cotes integration does not converge after an initial start of the analysis. Some oscillatory effects in the numerical predictions of the steady-state level are observed for a large number of integration points and low mesh density.

6.2 Numerical bridging results with finite stress value

Calculations were undertaken by method (a) with a finite stress value. The value of J_0 was included as described earlier. Two experimental tests were simulated: sizing A / epoxy and sizing A / polyester. As specific experiments were modelled, the data was not fitted to the average of all tests as shown in Table 1, but to the specific test data of some of the specimens (one for sizing A / epoxy, two for sizing A / polyester). For these specimens, we also recorded the crack length as a function of crack growth resistance.

The following data were used for the bridging law (for reference for the numerical nomenclature see Figure 11):

Sizing A / epoxy:
ΔJ_{ss}=3300 J/m^2
J_0=300 J/m^2
Δu_0 = 2 mm
Δu_1 = 0.05 mm $\rightarrow$ σ_0 = 27.65 MPa (σ_1 = 5.22 MPa)

Sizing A / polyester:
ΔJ_{ss}=4000 J/m^2
J_0=130 J/m^2
Δu_0 = 5.5 mm
Δu_1 = 0.05 mm $\rightarrow$ σ_0 = 16.64 MPa (σ_1 = 3.81 MPa)

The value for Δu_1 has to be chosen as small as possible if we want to imitate the original bridging law shape. However, there are numerical limitations, as with decreasing Δu_1 the value of σ_0 increases rapidly. The current value of Δu_1= 0.05 mm was found to lead to converging numerical studies with suitable levels of mesh refinement.

The fit with the crack growth resistance as shown in Figure 13 is satisfactory apart from slight differences in the initial increase of the crack growth resistance for small openings. For both materials, the model predicts a steeper increase than experimentally observed.

Figure 14 shows the comparison of the numerical predictions and the measured crack length for different values of crack growth resistance. The current numerical crack length is determined from a minimum positive node opening at the crack tip. An small threshold value of 1e-4 is used to determine the crack tip position. The results are quite insensitive to threshold value variations between 1e-4 and 1e-8.

It can be clearly seen that this comparison is much more sensitive to the bridging law shape, and the figure shows the deficiency of the current bridging law shape clearly. Numerically, for

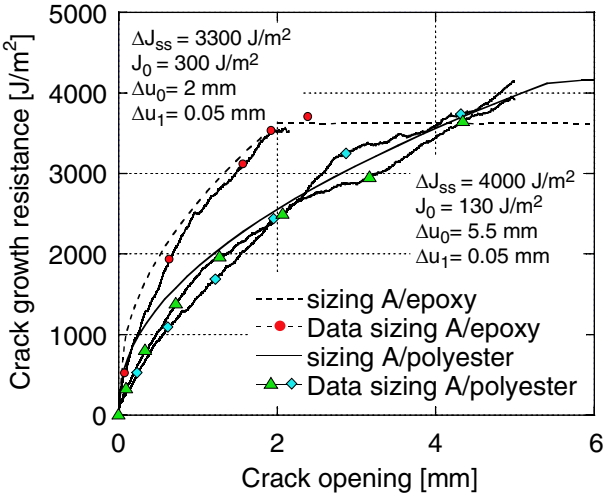


Figure 13. Comparison of crack growth resistance for method (a) with finite stress value

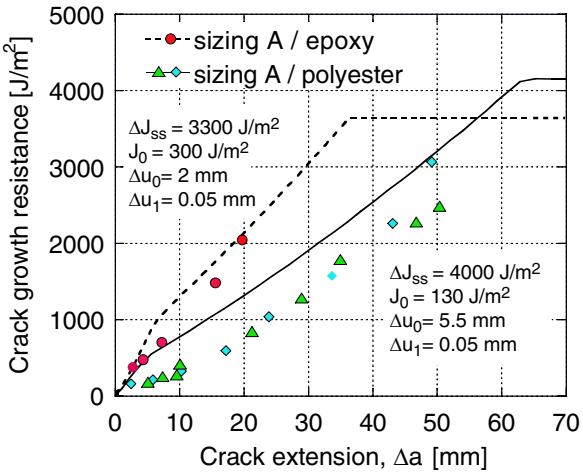


Figure 14. Comparison of crack growth development for method (a) with finite stress value

a given crack length a higher crack growth resistance is predicted.

The second concern is the initial prediction of crack growth. Naturally, due to the adjustment of the bridging law with respect to the incorporation of J_0 into the initial linear part, crack growth starts at zero crack growth resistance, and reaches the value after the initial crack opening of Δu_1= 0.05 mm. This corresponds to approximate crack lengths of 5 mm for epoxy, and 3 mm for the polyester as indicated by the ”knee” in Figure 14.

6.3 Numerical bridging results with power-law increase

We proposed splitting the values of J_0 and ΔJ_{ss} with this method as shown in Figure 12. Using an iterative procedure with Eq. (56), (57) and (55) as explained previously, we get the following input values for the bridging law:

Sizing A / epoxy:
Δu_0 = 2 mm
α = 100

$\Delta J_{ss} = 3300$ J/m^2
J_0=300 J/m^2 $\rightarrow \Delta u_2$ = 0.048 mm, σ_0 = 6.29 MPa and $\Delta J_{\text{ss, adj}}$=3906 J/m^2

Sizing A / polyester:
Δu_0 = 5.5 mm
α = 100
$\Delta J_{ss} = 4000$ J/m^2
J_0=130 J/m^2 $\rightarrow \Delta u_2$ = 0.021 mm, σ_0 = 6.29 MPa and $\Delta J_{\text{ss, adj}}$=4262 J/m^2

The determined values for Δu_2 are small in comparison with Δu_0 ($\frac{\Delta u_0}{\Delta u_2}$ >20). It was therefore decided to keep Δu_0 constant.

Figure 15 shows the comparison between numerical and experimental data for the crack growth resistance as a function of crack opening. It can be seen that the numerical predictions for small crack openings fit better with the splitted damage energies than it did with method (a).

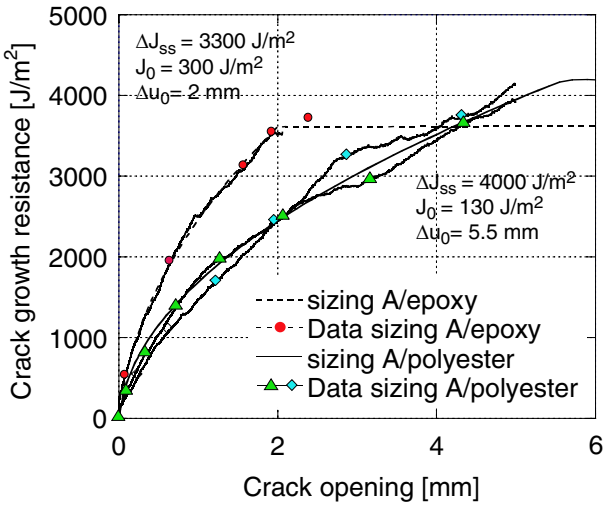


Figure 15. Comparison of crack growth resistance for method (b) with power-law increase

However, although the fit in Figure 15 seems nearly perfect with the proposed bridging law, the crack length as a function of the crack opening still shows deviations between the numerical and experimental results (see Figure 16). The current numerical crack length is determined from a minimum positive node opening at the crack tip, which is now defined as the value of Δu_2, and thereby corresponds to crack development after reaching the crack initiation energy of J_0. In the opinion of the author, this assumption has a physically more valid background than a randomly chosen small threshold value: one can consider damage models where fracture is caused by initial microvoids forming and growing together [15]. Upon reaching the value of damage initiation J_0, these microvoids will form a starting crack.

6.4 Adjustment of bridging law

The above examples showed that the bridging law shape can be further improved to provide a better fit with the experimental data; especially with respect to the crack length as a function of crack growth resistance. Previous trials with the above models were taken as the basis to improve the fit. Based on the above results, the following conclusions are drawn:

- Splitting of the bridging law into an initial fracture (J_0) and damage (ΔJ_{ss}) controlled part is of advantage based on the physical interpretation of the cohesive law parameters. For

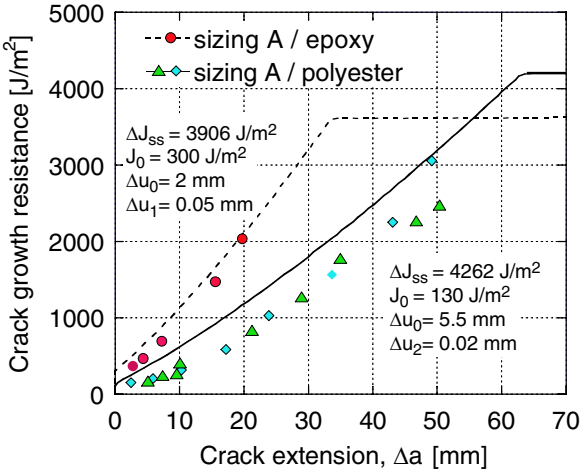


Figure 16. Comparison of crack growth development for method (b) with power-law increase

example, for a failure mode such as delamination between carbon / epoxy prepreg plies, the initial fracture energy surpasses the damage energy; while the bridging phenomenon of unidirectional glass fiber composites demonstrates the opposite material behaviour.

- Bridging law values should be derived from the experimental curves instead of being subject to a wide-ranging trial-and-error exercise. It is therefore desirable to estimate the value of Δu_2 from the experimental data directly. This would be even more important if Δu_2 is not small compared to Δu_0, i.e. both J_0 and ΔJ_{ss} have comparable order of magnitude.
- For method (b) splitting of initial fracture (J_0) and damage (ΔJ_{ss}) fracture energy improves the fit with the experimental data. However, the initial drop in the bridging law (for values larger than Δu_2) is still too severe; thereby resulting in an overprediction of the crack growth resistance for a certain crack length.

This changes to the bridging law shape are visualised in Figure 17. The shape is basically a combination of the previously discussed shapes for method (a) and (b). For K_{fac}=1 and Δu_1=0 we reset the bridging law to the shape of method (b). The derivation of these values is explained below.

Determination of Δu_2

Previously, the value for Δu_2 was derived with a nonlinear solving procedure. However, as the value actually has a physical meaning as explained in the last section, the experimental curves can be used to determine suitable values. This is shown in Figure 18.

The figure shows that prior to crack initiation, the experimental curves actually exhibit a small displacement of the order of 10-20 μm. Upon reaching J_0, these microvoids will start to form the starting crack. The corresponding crack growth resistance is the value of J_0 as used previously ($J_{0,\text{epoxy}}$=300 J/m^2, $J_{0,\text{poly}}$=130 J/m^2).

The corresponding displacement value can be used for Δu_2. Based on Eq. (55), we can then calculate a value for the required peak stress and thereby introduce a factor value K_{fac} in comparison to the original peak stress:

$$K_{\text{fac}} = \frac{J_0}{\Delta u_2 \sigma_0} \left(\frac{\alpha + 1}{\alpha} \right) \tag{59}$$

Determination of Δu_1

To improve the fit with the experimental data further, the decreasing bridging law slope after

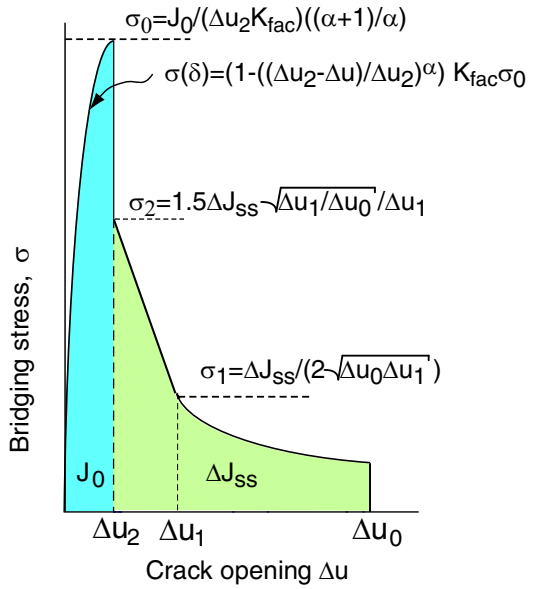


Figure 17. Adjusted bridging law

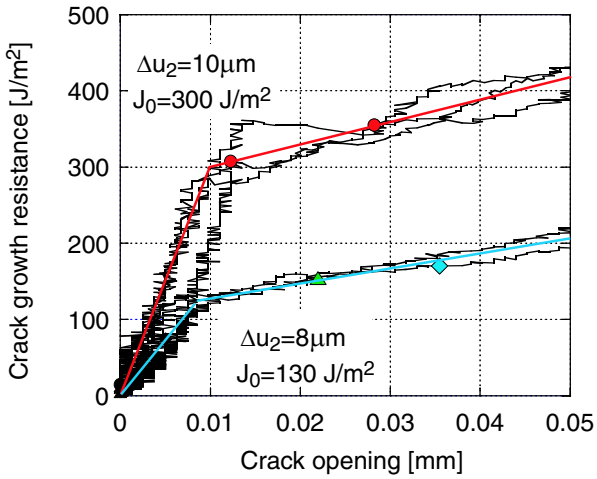


Figure 18. Determine Δu_2 experimentally

reaching Δu_2 needs to be reduced. We therefore take over the previously introduced linear decrease between Δu_2 and Δu_1 in this model.

This choice of an initially linear decrease is also based on previous numerical parameter studies of bridging law shapes [4, 11]. Bridging laws with linear decrease then show an s-shaped trend in the crack length development. Judging from the experimental data, where we overpredict more in the initial crack development stages, this type of model would improve the fit with the experimental data.

The value of Δu_1 now becomes the only adjustable value in the bridging law to be fitted to the experimental data points.

Determination of σ_0

σ_0 can be calculated from the requirement of total energy update equal to ΔJ_{ss} as previously with model (a), without including J_0 as this part of the fracture energy is taken care of during

the initial increase up to Δu_2.

$$\sigma_0 = \frac{\frac{3}{2}\Delta J_{\text{ss}} \left(\frac{\Delta u_1}{\delta_0}\right)^{\frac{1}{2}}}{\Delta u_1} \tag{60}$$

Bridging law parameters
J_0, ΔJ_{ss}, Δu_2 and Δu_0 can all be determined directly from the experimental data. The values are summarised in Table 3. Again, the value of Δu_2 is small in comparison with Δu_0, so there is no difference in the numerical prediction as to whether the length of the bridging law is given by Δu_0 or $\Delta u_0 + \Delta u_2$.

Table 3. Numerical values for the bridging law for different composite systems

Composite system	J_0 [J/m^2]	ΔJ_{ss} [J/m^2]	K_{fac}	Δu_0 [mm]	Δu_2 [mm]	Δu_1 [mm]
sizing A/epoxy	300	3300	8.0	2.0	0.01	1.0
sizing A/polyester	130	4000	9.0	5.5	0.008	2.5

Figure 19 shows the resulting fit for the crack growth resistance versus crack opening where the value of Δu_1 has been adjusted to 1.0 and 2.5 mm for epoxy and polyester, respectively. In both cases, this is about half the value of the total crack opening, Δu_0.

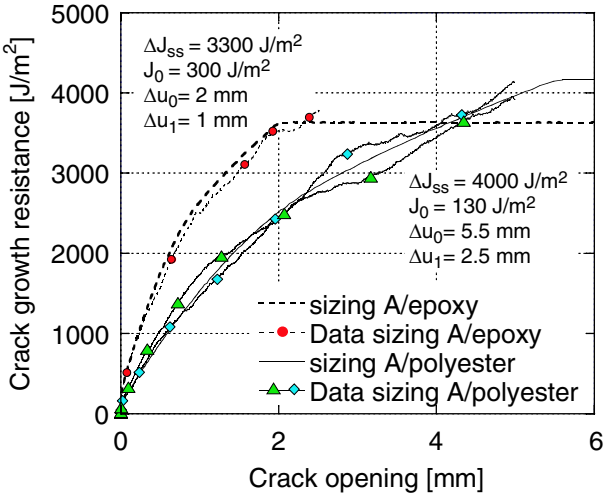


Figure 19. Comparison of crack growth resistance

Figure 20 now shows the excellent fit between the numerical and experimental data for the crack growth resistance versus crack length. The s-shaped crack length curve is now predicted extremely well. For a further check of the proposed crack bridging shape, the predicted crack profiles are studied together with the experimental predictions. This was undertaken for the polyester specimens as seen in Figure 21.

The figure shows the crack opening profile for a crack length of 50 mm. For accuracy of the experimental crack opening measurements, an opening of at least 0.2 mm must be given. It can be seen that the bridging law also in this case leads to a very good agreement between the two data sets. Additionally, the unbridged crack profile is shown in the figure. The bridging stresses across the crack faces lead - as expected - to a decrease in the opening of the crack.

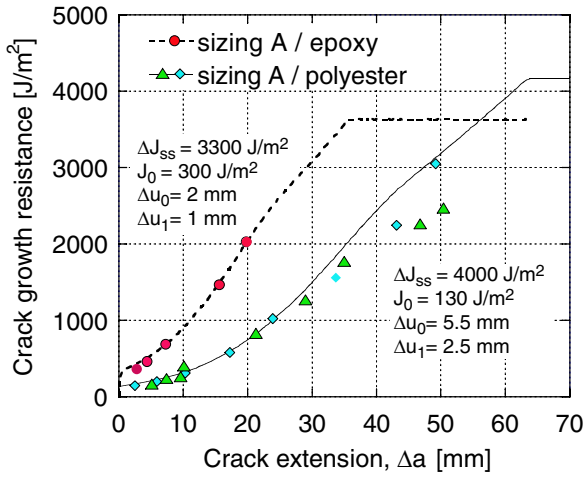


Figure 20. Comparison of crack growth development

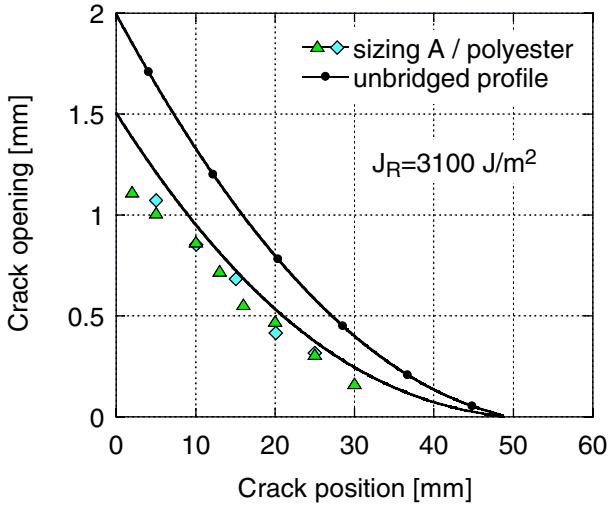


Figure 21. Comparison between numerical and experimental data for the crack opening profile for sizing A / polyester

The maximum opening at the crack notch is - for a length of 50 mm - reduced to 25% from 2.0 to 1.5 mm. This is quite a considerable effect of the bridging stresses.

Figure 22 shows the results from above together with the numerical predictions for sizing A/epoxy at the same crack growth resistance of J_R=3100 J/m^2. It can be seen that the crack opening and crack length for the same crack growth resistance are considerably smaller for sizing A/epoxy. Including the bridging stresses also changes the crack opening shape in the vicinity of the crack tip - however, this change is hard to determine correctly experimentally due to the small crack opening values.

Figure 23 now shows the numerical predictions for the crack opening profiles for both the sizing A/epoxy and sizing A/polyester system in comparison for the same crack opening. The curves are plotted for two different crack opening values of 0.3 and 1.2 mm. For these crack openings, sizing A/polyester gives the larger crack length together with a smaller crack growth resistance.

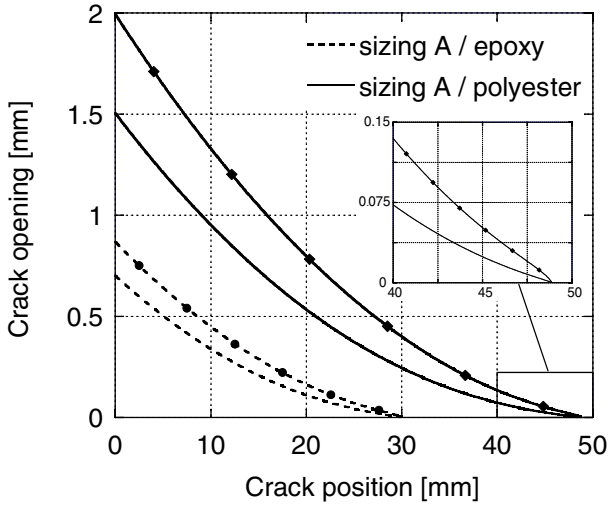


Figure 22. Comparison crack profiles for different bridging laws with J_R=3100 J/m^2

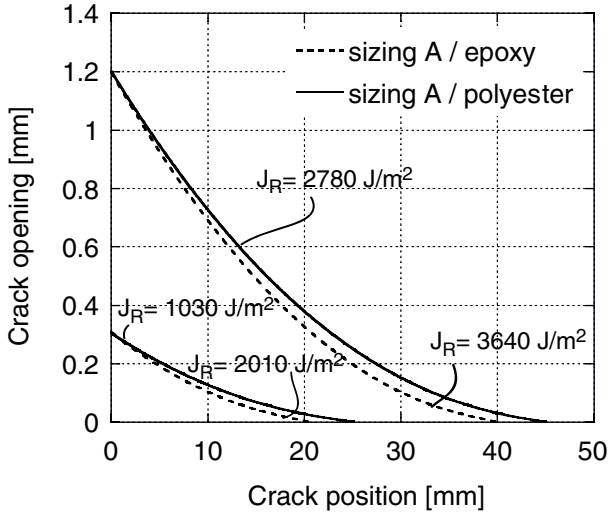


Figure 23. Comparison of crack profiles for different bridging laws

Unloading considerations

A possible procedure for unloading (i.e. crack closure during structural loading) was described earlier in Section 4.1 (see Figure 6). Although the current problem situation does not require an unloading procedure due to continually increasing crack opening, a theoretical consideration is given here. The splitting of the two fracture parts, J_0 and ΔJ_{ss}, can then also be used to change the history for elastic unloading: if we consider fracture initiation completed once the value of Δu_2 has been reached (void formation out of microvoids), elastic unloading to the origin should only occur for openings larger than Δu_2. For smaller values, the path can simply be reversed. This has not been investigated in detail as the current problem does not require unloading.

7 Summary

This report describes the implementation of a cohesive user element into the commercial FE code ABAQUS. This element type proves to be a versatile tool in predicting opening, crack length and crack profiles as a function of the crack growth resistance. In comparison with the earlier implemented UINTER contact routine, the cohesive element has the distinct advantage that a higher number of integration points can be used to improve the numerical performance for a given mesh density. This reduces the computational time for the analysis greatly.

The experimentally measured bridging law was optimised with the help of the numerical predictions. Experimental data such as crack length and crack profiles were found to give very good agreement with the adjusted law. The parameters for the new law can mostly be determined from the experiments; only one parameter remains to be fitted.

References

[1] J.R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.

[2] B.F. Sørensen and T.K. Jacobsen. Large scale bridging in composites: R-curves and bridging laws. *Composites: Part A*, 29A:1443–1451, 1998.

[3] S. Feih, J. Wei, P.K Kingshott, and B.F. Sørensen. The influence of fibre sizing on strength and fracture toughness of glass fibre reinforced composites. *Composites: Part A*, 36(2):245–255, 2005.

[4] Z. Suo, G. Bao, and B. Fan. Delamination R-curve phenomena due to damage. *Journal of Mechanics and Physics in Solids*, 1(40):1–16, 1992.

[5] B.F. Sørensen, K. Jørgensen, T.K. Jacobsen, and R.C. Østergaard. A general mixed mode fracture mechanics test specimen: The DCB-specimen loaded with uneven bending moments. Technical Report Risø-R-1394(EN), Risø National Laboratory, Roskilde, Denmark, March 2004.

[6] S.M. Spearing and A.G. Evans. The role of fibre bridging in the delamination resistance of fiber-reinforced composites. *Acta Metallica Materials*, 40(9):2191–2199, 1992.

[7] J.C.J. Schellekens and R. De Borst. On the numerical integration of interface elements. *International Journal for Numerical Methods in Engineering*, 36:43–66, 1993.

[8] C.G. Davila, P.P. Camanho, and M.F. de Moura. Mixed-mode decohesion elements for analyses of progressive delamination. In *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, number AIAA-01-1486, 2001.

[9] J. Chen, M. Crisfield, A.J. Kinloch, E.P. Busso, F.L. Matthews, and Y. Qiu. Predicting progressive delamination of composite material specimens via interface elements. *Mechanics of composite materials and structures*, 6:301–317, 1999.

[10] J. Segurada and J. LLorca. A new three-dimensional interface element to simulate fracture in composites. *International Journal of Solids and Structures*, 41:2977–2993, 2004.

[11] T.K. Jacobsen and B.F. Sørensen. Mode I intra-laminar crack growth in composites - modelling of R-curves measured from bridging laws. *Composites: Part A*, 32:1–11, 2001.

[12] Y. Mi, M.A. Crisfield, and G.A.O. Davies. Progressive delamination using interface elements. *Journal of Composite Materials*, 32:1246–1272, 1998.

[13] G. Alfano and M.A. Chrisfied. Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues. *International Journal for numerical methods in engineering*, 50:1701–1736, 2001.

[14] S. Feih. Modelling cohesive laws in finite element simulations via an adapted contact procedure in ABAQUS. Technical Report Risø-R-1463(EN), Risø National Laboratory, Roskilde, Denmark, July 2004.

[15] A. Cornec, I. Schneider, and K.H. Schwalbe. On the practical application of the cohesive model. *Engineering fracture mechanics*, 70:1963–1987, 2003.

A Implementation of the plane interface element

A.1 Derivation of force vector and stiffness matrix

The derivation of the force vector and stiffness matrix for the plane interface element are analogeous to the linear element presented earlier.

The plane interface element has 48 (3x16) degrees of freedom. The nodal displacements in the global coordinate system are given as:

$$\mathbf{d}_N = \left(d_x^1\, d_y^1\, d_z^1\, d_x^2\, d_y^2\, d_z^2 \ldots d_x^{16}\, d_y^{16}\, d_z^{16}\right)^T \tag{A.1}$$

$$\Delta\mathbf{u}_N = \mathbf{\Phi}\,\mathbf{d}_N = \left(-\mathbf{I}_{24\times 24} \;\;|\;\; \mathbf{I}_{24\times 24}\right)\mathbf{d}_N, \tag{A.2}$$

where $I_{24\times 24}$ denotes a unity matrix with 24 rows and columns.

Let $N_i(\xi,\eta)$ be the shape functions for the node pair i ($i = 1,\ldots,8$), where ξ and η stand for the local coordinates of the element with $-1 \le \xi \le 1$ and $-1 \le \eta \le 1$. The relative displacement between the nodes for each point within the element is then given by:

$$\Delta\mathbf{u}(\xi,\eta) = \begin{pmatrix} \Delta u_x(\xi\,,\eta) \\ \Delta u_y(\xi,\eta) \\ \Delta u_z(\xi,\eta) \end{pmatrix} = \mathbf{H}(\xi,\eta)\Delta\mathbf{u}_N, \tag{A.3}$$

where $H(\xi,\eta)$ is a 3x24 matrix containing the individual shape functions. It has the form

$$H(\xi,\eta) = \left(N_1(\xi,\eta)\,\mathbf{I}_{3x3} \;\;|\;\; N_2(\xi,\eta)\,\mathbf{I}_{3x3} \;\;|\;\; \cdots \;\; N_8(\xi,\eta)\,\mathbf{I}_{3x3}\right). \tag{A.4}$$

for plane interface elements.

As a result, we get

$$\Delta\mathbf{u}(\xi,\eta) = \mathbf{H}\mathbf{\Phi}\mathbf{d}_N = \mathbf{B}\mathbf{d}_N, \tag{A.5}$$

where $\mathbf{B}$ is of the dimension 3x48 and $\Delta\mathbf{u}$ of the dimension 3x1.

The coordinates of any specific reference position can be derived according to Eq. (A.3):

$$\mathbf{x}^R(\xi,\eta) = \mathbf{H}(\xi,\eta)\mathbf{x}_N^R \tag{A.6}$$

The tangential plane is spanned by two vectors, $\mathbf{v}_\xi$ and $\mathbf{v}_\eta$. They are, at a given point, obtained by differentiating the global position vector with respect to the local coordinates. Although $\mathbf{v}_\xi$ and $\mathbf{v}_\eta$ are generally not orthogonal to each other, their vector product defines the surface normal. Therefore, the local normal vector is obtained by:

$$\mathbf{t}_n = \frac{1}{||\frac{\partial\mathbf{x}^R}{\partial\xi} \times \frac{\partial\mathbf{x}^R}{\partial\eta}||}\left(\frac{\partial\mathbf{x}^R}{\partial\xi} \times \frac{\partial\mathbf{x}^R}{\partial\eta}\right) \tag{A.7}$$

The tangential coordinates are defined as:

$$\mathbf{t}_1 = \frac{1}{||\frac{\partial\mathbf{x}^R}{\partial\xi}||}\frac{\partial\mathbf{x}^R}{\partial\xi} \tag{A.8}$$

$$\mathbf{t}_2 = \mathbf{t}_n \times \mathbf{t}_1 \tag{A.9}$$

The components $\mathbf{t}_1, \mathbf{t}_2$ and $\mathbf{t}_n$ represent the direction cosines of the local coordinate system to the global one, thus defining the 3×3 transformation tensor $\mathbf{\Theta}$:

$$\mathbf{\Theta} = \left(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_n\right). \tag{A.10}$$

The local displacements are then obtained by

$$\Delta\mathbf{u}_{\text{loc}} = \mathbf{\Theta}^T\Delta\mathbf{u}. \tag{A.11}$$

The element stiffness matrix and the right hand side nodal force vector are required for the UEL subroutine in ABAQUS.

The nodal force vector is defined as

$$\mathbf{f}_N^{\text{el}} = \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T \mathbf{\Theta}^T \mathbf{t}_{\text{loc}} \det J \, d\xi d\eta, \tag{A.12}$$

where $\mathbf{t}_{\text{loc}}$ is the $3x1$ vector defining the bridging stresses. The bridging stresses are connected to the opening displacements by the cohesive law as derived previously. It should be noticed that - for numerical purposes - the relationship can be expressed in various forms (analytical, piecewise linear, etc).

$\det J$ is the Jacobian defined by the transformation of the current element coordinates (ξ, η) to the global coordinates (x, y, z), and defined as

$$\det J = \sqrt{(\det J1)^2 + (\det J2)^2 + (\det J3)^3} \quad \text{with} \tag{A.13}$$

$$\det J1 = \partial x_\xi^R \partial y_\eta^R - \partial x_\eta^R \partial y_\xi^R \tag{A.14}$$

$$\det J2 = \partial x_\xi^R \partial z_\eta^R - \partial x_\eta^R \partial z_\xi^R \tag{A.15}$$

$$\det J3 = \partial y_\xi^R \partial z_\eta^R - \partial y_\eta^R \partial z_\xi^R \tag{A.16}$$

Note that depending on the orientation of the local to the global coordinate system, the components $\det J1, \det J2$ and $\det J3$ can be negative.

The tangent stiffness matrix (note sign convention for ABAQUS) is defined as

$$\mathbf{K}^{\text{el}} = -\frac{\partial \mathbf{f}_N^{\text{el}}}{\partial \mathbf{d}^{\text{el}}} = -\int_{\text{el}} \mathbf{B}^T \mathbf{\Theta}^T \mathbf{D}_{\text{loc}} \mathbf{\Theta} \mathbf{B} \, dS_{\text{el}} \tag{A.17}$$

$$\mathbf{K}_{48\times 48} = -\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T \mathbf{\Theta}^T \mathbf{D}_{\text{loc}} \mathbf{\Theta} \mathbf{B} \, d\xi d\eta \tag{A.18}$$

A.2 State variables for the analysis

The number of state variables for each element depends on the chosen integration procedure. At each integration point of the plane element, there are three state variables for the relative displacement (x-, y- and z-direction) and the local stresses acting in the x-, y- and z-direction. Furthermore, there is one value describing the current interaction state between the two surfaces for the purpose of stability analysis and position tracking during the numerical procedure. For a standard integration procedure with 9 integration points, this results in 63 (9x7) state variables for the plane element. As ABAQUS will not terminate the analysis if an insufficient number of state variables is provided for the chosen integration procedure, this user input error is checked within the UEL subroutine.

B Quadratic shape functions and derivatives

B.1 Line element

The three quadratic shape functions for line element are as follows:

$$N_1(\xi) = \frac{1}{2}(-\xi + \xi^2) \tag{B.1}$$

$$N_2(\xi) = \frac{1}{2}(\xi + \xi^2) \tag{B.2}$$

$$N_3(\xi) = 1 - \xi^2 \tag{B.3}$$

Derivative of shape functions

$$N_{1,\xi}(\xi) = -\frac{1}{2} + \xi \tag{B.4}$$

$$N_{2,\xi}(\xi) = \frac{1}{2} + \xi \tag{B.5}$$

$$N_{3,\xi}(\xi) = -2\xi \tag{B.6}$$

B.2 Plane element

The quadratic shape functions for the plane element are given below:

$$N_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) - \frac{1}{2}(N_5 + N_8) \tag{B.7}$$

$$N_2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta) - \frac{1}{2}(N_5 + N_6) \tag{B.8}$$

$$N_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta) - \frac{1}{2}(N_6 + N_7) \tag{B.9}$$

$$N_4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta) - \frac{1}{2}(N_7 + N_8) \tag{B.10}$$

$$N_5(\xi,\eta) = \frac{1}{2}(1-\xi^2)(1-\eta) \tag{B.11}$$

$$N_6(\xi,\eta) = \frac{1}{2}(1+\xi)(1-\eta^2) \tag{B.12}$$

$$N_7(\xi,\eta) = \frac{1}{2}(1-\xi^2)(1+\eta) \tag{B.13}$$

$$N_8(\xi,\eta) = \frac{1}{2}(1-\xi)(1-\eta^2) \tag{B.14}$$

Derivative of shape functions

$$N_{1,\xi} = -\frac{1}{4}(1-\eta) - \frac{1}{2}(N_{5,\xi} + N_{8,\xi}) \tag{B.15}$$

$$N_{2,\xi} = \frac{1}{4}(1-\eta) - \frac{1}{2}(N_{5,\xi} + N_{6,\xi}) \tag{B.16}$$

$$N_{3,\xi} = \frac{1}{4}(1+\eta) - \frac{1}{2}(N_{6,\xi} + N_{7,\xi}) \tag{B.17}$$

$$N_{4,\xi} = -\frac{1}{4}(1+\eta) - \frac{1}{2}(N_{7,\xi} + N_{8,\xi}) \tag{B.18}$$

$$N_{5,\xi} = -\xi(1-\eta) \tag{B.19}$$

$$N_{6,\xi} = \frac{1}{2}(1-\eta^2) \tag{B.20}$$

$$N_{7,\xi} = -\xi(1+\eta) \tag{B.21}$$

$$N_{8,\xi} = -\frac{1}{2}(1-\eta^2) \tag{B.22}$$

$$N_{1,\eta} = -\frac{1}{4}(1-\xi) - \frac{1}{2}(N_{5,\eta} + N_{8,\eta}) \tag{B.23}$$

$$N_{2,\eta} = -\frac{1}{4}(1+\xi) - \frac{1}{2}(N_{5,\eta} + N_{6,\eta}) \tag{B.24}$$

$$N_{3,\eta} = \frac{1}{4}(1+\xi) - \frac{1}{2}(N_{6,\eta} + N_{7,\eta}) \tag{B.25}$$

$$N_{4,\eta} = \frac{1}{4}(1-\xi) - \frac{1}{2}(N_{7,\eta} + N_{8,\eta}) \tag{B.26}$$

$$N_{5,\eta} = -\frac{1}{2}(1-\xi^2) \tag{B.27}$$

$$N_{6,\eta} = -\eta(1+\xi) \tag{B.28}$$

$$N_{7,\eta} = \frac{1}{2}(1-\xi^2) \tag{B.29}$$

$$N_{8,\eta} = -\eta(1-\xi) \tag{B.30}$$

C Integration points and weights

C.1 Gauss integration point positions and weights

3 points

$$
\begin{aligned}
p_1 &= -\sqrt{0.6} & w_1 &= 0.555 \\
p_2 &= 0.0 & w_2 &= 0.888 \\
p_3 &= \sqrt{0.6} & w_3 &= 0.555
\end{aligned}
$$

6 points

$$
\begin{aligned}
p_1 &= -0.9324695142031520 & w_1 &= 0.1713244923791709 \\
p_2 &= -0.6612093864662646 & w_2 &= 0.3607615730481379 \\
p_3 &= -0.2386191860831968 & w_3 &= 0.4679139345726913 \\
p_4 &= 0.2386191860831968 & w_4 &= 0.4679139345726913 \\
p_5 &= 0.6612093864662646 & w_5 &= 0.3607615730481379 \\
p_6 &= 0.9324695142031520 & w_6 &= 0.1713244923791709
\end{aligned}
$$

12 points

$$
\begin{aligned}
p_1 &= -0.981560634246732 & w_1 &= 0.04717533638647547 \\
p_2 &= -0.904117256370452 & w_2 &= 0.1069393259953637 \\
p_3 &= -0.7699026741943177 & w_3 &= 0.1600783285433586 \\
p_4 &= -0.5873179542866143 & w_4 &= 0.2031674267230672 \\
p_5 &= -0.3678314989981804 & w_5 &= 0.2334925365383534 \\
p_6 &= -0.12523340851114688 & w_6 &= 0.2491470458134027 \\
p_7 &= 0.12523340851114688 & w_7 &= 0.2491470458134027 \\
p_8 &= 0.3678314989981804 & w_8 &= 0.2334925365383534 \\
p_9 &= 0.5873179542866143 & w_9 &= 0.2031674267230672 \\
p_{10} &= 0.7699026741943177 & w_{10} &= 0.1600783285433586 \\
p_{11} &= 0.904117256370452 & w_{11} &= 0.1069393259953637 \\
p_{12} &= 0.981560634246732 & w_{12} &= 0.04717533638647547
\end{aligned}
$$

C.2 Newton-Cotes integration point positions and weights

3 points

$$
\begin{aligned}
p_1 &= -1.0 & w_1 &= 0.333 \\
p_2 &= 0.0 & w_2 &= 1.333 \\
p_3 &= 1.0 & w_3 &= 0.333
\end{aligned}
$$

6 points

$$
\begin{aligned}
p_1 &= -1.0 & w_1 &= 19/144 \\
p_2 &= -3/5 & w_2 &= 75/144 \\
p_3 &= -1/5 & w_3 &= 50/144 \\
p_4 &= 1/5 & w_4 &= 50/144 \\
p_5 &= 3/5 & w_5 &= 75/144 \\
p_6 &= 1.0 & w_6 &= 19/144
\end{aligned}
$$

12 points

$$
\begin{aligned}
p_1 &= -1.0 & w_1 &= 2171465/43545600 \\
p_2 &= -9/11 & w_2 &= 13486539/43545600 \\
p_3 &= -7/11 & w_3 &= -3237113/43545600 \\
p_4 &= -5/11 & w_4 &= 25226685/43545600 \\
p_5 &= -3/11 & w_5 &= -9595542/43545600 \\
p_6 &= -1/11 & w_6 &= 15493566/43545600 \\
p_7 &= 1/11 & w_7 &= 15493566/43545600 \\
p_8 &= 3/11 & w_8 &= -9595542/43545600 \\
p_9 &= 5/11 & w_9 &= 25226685/43545600 \\
p_{10} &= 7/11 & w_{10} &= -3237113/43545600 \\
p_{11} &= 9/11 & w_{11} &= 13486539/43545600 \\
p_{12} &= 1.0 & w_{12} &= 2171465/43545600
\end{aligned}
$$

D ABAQUS coding for the quadratic line element

The following code will be compiled and linked when calling the ABAQUS job. For unix systems, the command prompt will be as follows:

```
abaqus job=jobname user=interface
```

This syntax is of course dependent on the platform you are using. For further information see the ABAQUS manuals. The following code for the user subroutine UEL is written in FORTRAN 77 and saved as 'interface.f'.

```
      SUBROUTINE UEL(RHS,AMATRX,SVARS,ENERGY,NDOFEL,NRHS,NSVARS,
     1 PROPS,NPROPS,COORDS,MCRD,NNODE,U,DU,V,A,JTYPE,TIME,DTIME,
     2 KSTEP,KINC,JELEM,PARAMS,NDLOAD,JDLTYP,ADLMAG,PREDEF,
     3 NPREDF,LFLAGS,MLVARX,DDLMAG,MDLOAD,PNEWDT,JPROPS,NJPROP,
     4 PERIOD)
C
      INCLUDE 'ABA_PARAM.INC'
      PARAMETER (ZERO = 0.D0, HALF=0.5D0, ONE= 1.0D0, TWO=2.0d0,
     1     THREE= 3.0d0, TOL=-1E-5)

      DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),
     1 SVARS(NSVARS),ENERGY(8),PROPS(*),COORDS(MCRD,NNODE),
     2 U(NDOFEL),DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),
     3 PARAMS(3),JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),
     4 DDLMAG(MDLOAD,*),PREDEF(2,NPREDF,NNODE),LFLAGS(*),
     5 JPROPS(*)

C     GENERAL ELEMENT VALUES
      DIMENSION STRESS(MCRD)
      DIMENSION DDSDDR(MCRD,MCRD)

C     GAUSS INTEGRATION VARIABLES (3 INTEG POINT)
      DIMENSION GAUSS3(3), WEIGHT3(3), COTNEW(3), CWEIGHT(3)
      DIMENSION GAUSS6(6), WEIGHT6(6), COTNEW6(6), CW6(6)
      DIMENSION GAUSS12(12), WEIGHT12(12), COTNEW12(12), CW12(12)

C     ARRAYS FOR QUADRATIC LINE ELEMENT
      DIMENSION DNDXI(3), DELTA_U(6), DU_CONT(MCRD), DU_LOC(MCRD)
      DIMENSION H(MCRD,6), C_COOR(MCRD,NNODE), PSI(6,NDOFEL)
      DIMENSION B(MCRD, NDOFEL), BT(NDOFEL, MCRD)
      DIMENSION A1(NDOFEL, MCRD), A2(NDOFEL, NDOFEL)
      DIMENSION AV_COOR(MCRD, 3), V_XI(MCRD), V_N(MCRD)
      DIMENSION THETA(MCRD, MCRD), STR_GLOB(MCRD)
      DIMENSION D_GLOB(MCRD, MCRD), DD1(MCRD, MCRD)

      data iuel/0/
      save iuel


C
C     QUADRATIC LINE ELEMENT
C     SVARS  - In 1, contains the lOpenClose identifier
C            - In 3-4, contains the traction stiffness
C            - In 5-6, contains the traction opening
C
C     INITIALISATION: IMPORTANT!! FORTRAN DOES NOT PUT ZEROS IN THERE AUTOMATICALLY
      CALL KASET2(AMATRX, NDOFEL, NDOFEL)

      IF (NHRS.EQ.1) THEN
         CALL KASET1(RHS, MLVARX)
      ELSE
         CALL KASET2(RHS, MLVARX, NRHS)
      END IF

      CALL KASET2(PSI, 6, NDOFEL)
```

```
      CALL KASET2(H, MCRD, 6)
      CALL KASET2(AV_COOR, MCRD, 3)
      CALL KASET1(V_XI, MCRD)
      CALL KASET1(V_N, MCRD)
      CALL KASET2(THETA, MCRD, MCRD)

      CALL KASET2(DDSDDR, MCRD, MCRD)
      CALL KASET2(D_GLOB, MCRD, MCRD)

      CALL KASET1(STRESS, MCRD)
      CALL KASET1(STR_GLOB, MCRD)


C     REAL INPUT PROPERTIES
      WIDTH  = PROPS(7)    ! Width of elements (same as solid section width for solid elements)

C     INTEGER INPUT PROPERTIES
      NINTP  = JPROPS(1)  ! Number of integration points
      INTS   = JPROPS(2)  ! Integration point scheme (1: gauss, 2: newton cotes)

C     INFORMATION OUTPUT AND CHECK
      IF (iuel.EQ.0) THEN

         OPEN(15,FILE=
     1        '/user/feih/abaqus/uel/verify.out')


         write(7,*) 'First call to UEL-----------------'

         WRITE(7,*) 'DEGREES OF FREEDOM:',NDOFEL
         write(7,*) 'number of nodes:', NNODE
         write(7,*) 'number of integration points:', NINTP
         write(7,*) 'Integration scheme:', INTS
         write(7,*) 'maximum coords:', MCRD
         write(7,*) 'number of variables:', NSVARS
         write(7,*) 'number of real properties', NPROPS
         write(7,*) 'number of integer properties', NJPROP
         write(7,*) 'dimensioning parameter:', MLVARX
         write(7,*) 'KINC:', KINC


         write(7,*) 'LFLAGS(1)=', LFLAGS(1)
         write(7,*) 'LFLAGS(2)=', LFLAGS(2)
         write(7,*) 'LFLAGS(3)=', LFLAGS(3)
         write(7,*) 'LFLAGS(4)=', LFLAGS(4)
         write(7,*) 'LFLAGS(5)=', LFLAGS(5)

C     CHECKING FOR THE RIGHT NUMBER OF NODES
         IF (NNODE.NE.6) THEN
            CALL STDB_ABQERR(-3, '6 nodes required for interface element:
     1              specified number of nodes is incorrect',0,0.0,' ')
         END IF

C     Checking for number of state variables
         minnum = NINTP*5
         IF (NSVARS.LT.minnum) THEN
            CALL STDB_ABQERR(-3, 'Number of state variables too small for
     1              chosen number of integration points!',MINNUM,0.0,' ')
         END IF

         IUEL = 1

      END IF


C      WRITE(7,*) 'New call to UEL'
C      do k=1, NDOFEL
C         write(7,*) 'U', U(k)
C      end do
```

```
      call flush_(15)


C     Calculate relation matrix
C     Calculate relative opening at node pairs
C     DEFINE DELTA_U=U_TOP - U_BOTTOM

      DO 10 K = 1, NDOFEL/2

         PSI(K, K) = -ONE
         PSI(K, K+NDOFEL/2) = ONE

  10  END DO

C     Compute nodal coordinates in deformed state
C     ADD PROPER COORDINATE TRANSFORMATION LATER
      DO 20 I=1,MCRD
         DO 30 J=1, NNODE

            NN=I+(J-1)*MCRD
            C_COOR(I,J)=COORDS(I,J) + U(NN)

  30     END DO
  20  END DO

c     Reference coordinate system (midpoint averages)
      DO 31 I=1, MCRD
         DO 32 J=1, NNODE/2

            AV_COOR(I,J)=ONE/TWO*(C_COOR(I,J)+C_COOR(I,J+NNODE/2))

 32      END DO
 31   END DO


c     Gaussian integration (3 gauss points)
      GAUSS3(1) =  -SQRT(0.6)
      GAUSS3(2) =  ZERO
      GAUSS3(3) =  SQRT(0.6)

      WEIGHT3(1) = 0.55555555555555
      WEIGHT3(2) = 0.88888888888888
      WEIGHT3(3) = 0.55555555555555

c     Gaussian integration (6 gauss points)
      GAUSS6(1) = -0.932469514203152
      GAUSS6(2) = -0.6612093864662646
      GAUSS6(3) = -0.2386191860831968
      GAUSS6(4) =  0.2386191860831968
      GAUSS6(5) =  0.6612093864662646
      GAUSS6(6) =  0.932469514203152

      WEIGHT6(1) = 0.1713244923791709
      WEIGHT6(2) = 0.3607615730481379
      WEIGHT6(3) = 0.4679139345726913
      WEIGHT6(4) = 0.4679139345726913
      WEIGHT6(5) = 0.3607615730481379
      WEIGHT6(6) = 0.1713244923791709

c     Gaussian integration (12 gauss points)
      GAUSS12(1) = -0.981560634246732
      GAUSS12(2) = -0.904117256370452
      GAUSS12(3) = -0.7699026741943177
      GAUSS12(4) = -0.5873179542866143
      GAUSS12(5) = -0.3678314989981804
      GAUSS12(6) = -0.12523340851114688
      GAUSS12(7) =  0.12523340851114688
      GAUSS12(8) =  0.3678314989981804
```

```
      GAUSS12(9) =  0.5873179542866143
      GAUSS12(10) = 0.7699026741943177
      GAUSS12(11) = 0.904117256370452
      GAUSS12(12) = 0.981560634246732

      WEIGHT12(1) = 0.04717533638647547
      WEIGHT12(2) = 0.1069393259953637
      WEIGHT12(3) = 0.1600783285433586
      WEIGHT12(4) = 0.2031674267230672
      WEIGHT12(5) = 0.2334925365383534
      WEIGHT12(6) = 0.2491470458134027
      WEIGHT12(7) = 0.2491470458134027
      WEIGHT12(8) = 0.2334925365383534
      WEIGHT12(9) = 0.2031674267230672
      WEIGHT12(10) = 0.1600783285433586
      WEIGHT12(11) = 0.1069393259953637
      WEIGHT12(12) = 0.04717533638647547

c     Newton Cotes integration (3 integration points)
      COTNEW(1) =  -ONE
      COTNEW(2) =  ZERO
      COTNEW(3) =   ONE

      CWEIGHT(1) = ONE/THREE
      CWEIGHT(2) = ONE + ONE/THREE
      CWEIGHT(3) = ONE/THREE

c     Newton Cotes integration (6 integration points)
      COTNEW6(1) =  -ONE
      COTNEW6(2) =  -3.0d0/5.0d0
      COTNEW6(3) =  -1.0d0/5.0d0
      COTNEW6(4) =  1.0d0/5.0d0
      COTNEW6(5) =  3.0d0/5.0d0
      COTNEW6(6) =   ONE

      CW6(1) = 19.0d0/144.0d0
      CW6(2) = 75.0d0/144.0d0
      CW6(3) = 50.0d0/144.0d0
      CW6(4) = 50.0d0/144.0d0
      CW6(5) = 75.0d0/144.0d0
      CW6(6) = 19.0d0/144.0d0

c     Newton Cotes integration (12 integration points)
      COTNEW12(1) =  -ONE
      COTNEW12(2) =  -9.0d0/11.0d0
      COTNEW12(3) =  -7.0d0/11.0d0
      COTNEW12(4) =  -5.0d0/11.0d0
      COTNEW12(5) =  -3.0d0/11.0d0
      COTNEW12(6) =  -1.0d0/11.0d0
      COTNEW12(7) =  1.0d0/11.0d0
      COTNEW12(8) =  3.0d0/11.0d0
      COTNEW12(9) =  5.0d0/11.0d0
      COTNEW12(10) = 7.0d0/11.0d0
      COTNEW12(11) = 9.0d0/11.0d0
      COTNEW12(12) = ONE


      CW12(1) = 2171465.0d0/43545600.0d0
      CW12(2) = 13486539.0d0/43545600.0d0
      CW12(3) = -3237113.0d0/43545600.0d0
      CW12(4) = 25226685.0d0/43545600.0d0
      CW12(5) = -9595542.0d0/43545600.0d0
      CW12(6) = 15493566.0d0/43545600.0d0
      CW12(7) = 15493566 .0d0/43545600.0d0
      CW12(8) = -9595542.0d0/43545600.0d0
      CW12(9) = 25226685.0d0/43545600.0d0
      CW12(10) = -3237113.0d0/43545600.0d0
      CW12(11) = 13486539.0d0/43545600.0d0
      CW12(12) = 2171465.0d0/43545600.0d0
```

```
C
      IF (LFLAGS(3).EQ.1) THEN

C        Normal incrementation (RHS and AMATRX required)

         IF (LFLAGS(1).EQ.1.OR.LFLAGS(1).EQ.2) THEN
C           *STATIC AND *STATIC, DIRECT

C           LOOP OVER INTEGRATION POINTS
            DO 100 IINTP = 1,NINTP

               IF (NINTP.EQ.3.AND.INTS.EQ.1) THEN

                  POINT = GAUSS3(IINTP)
                  WEIGHT = WEIGHT3(IINTP)

               ELSE IF (NINTP.EQ.6.AND.INTS.EQ.1) THEN

                  POINT = GAUSS6(IINTP)
                  WEIGHT = WEIGHT6(IINTP)

               ELSE IF (NINTP.EQ.12.AND.INTS.EQ.1) THEN

                  POINT = GAUSS12(IINTP)
                  WEIGHT = WEIGHT12(IINTP)

               ELSE IF (NINTP.EQ.3.AND.INTS.EQ.2) THEN

                  POINT = COTNEW(IINTP)
                  WEIGHT = CWEIGHT(IINTP)

               ELSE IF (NINTP.EQ.6.AND.INTS.EQ.2) THEN

                  POINT = COTNEW6(IINTP)
                  WEIGHT = CW6(IINTP)

               ELSE IF (NINTP.EQ.12.AND.INTS.EQ.2) THEN

                  POINT = COTNEW12(IINTP)
                  WEIGHT = CW12(IINTP)

               ELSE

                  WRITE(7,*) 'Unspecified integration required'
                  CALL FLUSH_(7)
                  CALL XIT

               END IF

C              Shape function value
               H1 = ONE/TWO*(-POINT + POINT**TWO)
               H2 = ONE/TWO*( POINT + POINT**TWO)
               H3 = ONE - POINT**TWO

C              DERIVATIVE OF SHAPE FUNCTION VALUE (3X1 MATRIX)
               DNDXI(1) = -ONE/TWO + POINT
               DNDXI(2) =  ONE/TWO + POINT
               DNDXI(3) = -TWO*POINT

C              H matrix
               H(1,1) = H1
               H(2,2) = H1
               H(1,3) = H2
               H(2,4) = H2
               H(1,5) = H3
               H(2,6) = H3

c               write(7,*) 'Starting loop over integration points'
```

```
c               write(7,*) 'INTP POINT and WEIGHT', IINTP, POINT, WEIGHT
C               call flush_(7)

                CALL KASET2(B, MCRD, NDOFEL)

                DO 110 I=1, MCRD
                   DO 120 J=1, NDOFEL
                      DO 130 K=1, NDOFEL/2

                         B(I,J) = B(I,J) + H(I,K)*PSI(K,J)

 130                  END DO
 120               END DO
 110             END DO

C               TRANSPOSED B MATRIX
                DO 140 I=1, MCRD
                   DO 150 J=1, NDOFEL

                      BT(J,I) = B(I,J)

 150               END DO
 140             END DO

C               CALCULATE GLOBAL DISPLACEMENT AT INTEGRATION POINT
C               FROM CONTINUOUS DISPLACEMENT

                CALL KASET1(DU_CONT, MCRD)

                DO 160 I=1, MCRD
                   DO 170 J=1, NDOFEL

                      DU_CONT(I) = DU_CONT(I) + B(I,J)*U(J)

 170               END DO
 160             END DO


C               LOCAL COORDINATE SYSTEM
C               (USE AVERAGE OF DEFORMED X-POSITIONS OF TOP AND BOTTOM)

                X_xi = ZERO
                Y_xi = ZERO

                DO 180 I=1,3
                      X_xi = X_xi +
     1                       DNDXI(I)*AV_COOR(1,I)

                      Y_xi = Y_xi +
     1                       DNDXI(I)*AV_COOR(2,I)
 180             END DO

c               Jacobian (vector length in xi-direction)
                DETJ = sqrt(X_xi**TWO + Y_xi**TWO)

                IF (DETJ.LT.ZERO) THEN
                   write(7,*) 'Negative Jacobian encountered!
     1             Check element and nodal definition for elem', JELEM
                   CALL XIT
                END IF

C               Local coordinate vector
                V_XI(1) = X_XI/DETJ
                V_XI(2) = Y_XI/DETJ

C               Normal vector in 90 degree angle
                V_N(1) = - V_XI(2)
                V_N(2) = V_XI(1)
```

```
c              Rotational matrix
               THETA(1,1) = V_XI(1)
               THETA(2,1) = V_XI(2)
               THETA(1,2) = V_N(1)
               THETA(2,2) = V_N(2)

c              Relative displacement in local coordinate system

               CALL KASET1(DU_LOC, MCRD)
               DO 181 I=1, MCRD
                  DO 182 J=1, MCRD

                     DU_LOC(I) = DU_LOC(I) + THETA(J,I)*DU_CONT(J)

 182              END DO
 181           END DO

c              over-closure check (can be used as re-start criterion - see uinter)
               IF (DU_LOC(2).LT.TOL) THEN
                  write(7,*) 'Over-closure at element', JELEM
               END IF

C               write (7,*) 'DU_LOC:', DU_LOC(1), DU_LOC(2), IINTP
C               CALL FLUSH_(7)

C              CALCULATE STRESS AND TRACTION STIFFNESS BASED ON RELATIVE DISPLACEMENT
               CALL KTRACN(DU_LOC, PROPS, STRESS, DDSDDR,
     1                     MCRD, SVARS, NSVARS, IINTP, NINTP, KINC, JELEM)


c              dummy stiffness for friction (no influence under mode I opening when coupled with equation)
c              for accuracy there should be coupling terms,
c              but again: no influence under mode I opening

               DDSDDR(1,1) = 10000

C              RHS ASSEMBLY
C              CHECK FOR APPLIED LOADS ON STRUCTURE

               IF (NDLOAD.NE.0) THEN
                  WRITE(7,*) 'Element loads not implemented'
                  CALL FLUSH_(7)
                  CALL XIT
               END IF

C              Stiffness matrix
c              Transformation

               CALL KASET2(DD1, MCRD, MCRD)
               DO 183 I=1, MCRD
                  DO 184 J=1, MCRD
                     DO 185 K=1, MCRD

                        DD1(I,J) = DD1(I,J) + DDSDDR(I,K)*THETA(J,K)

 185                 END DO
 184              END DO
 183           END DO

               CALL KASET2(D_GLOB, MCRD, MCRD)
               DO 186 I=1, MCRD
                  DO 187 J=1, MCRD
                     DO 188 K=1, MCRD

                        D_GLOB(I,J) = D_GLOB(I,J) + THETA(I,K)*DD1(K,J)

 188                 END DO
 187              END DO
 186           END DO
```

```
              CALL KASET2 (A1, NDOFEL, MCRD)
              DO 190 I=1, NDOFEL
                 DO 191 J=1, MCRD
                    DO 192 K=1, MCRD

                          A1(I,J) = A1(I,J) + BT(I,K)*D_GLOB(K,J)

 192                   END DO
 191                 END DO
 190           END DO


              CALL KASET2 (A2, NDOFEL, NDOFEL)
              DO 195 I=1, NDOFEL
                 DO 196 J=1, NDOFEL
                    DO 197 K=1, MCRD

                          A2(I,J) = A2(I,J) + A1(I,K)*B(K,J)

 197                   END DO
 196                 END DO
 195              END DO



              DO 200 I=1, NDOFEL
                 DO 201 J=1, NDOFEL

                    AMATRX(I,J) = AMATRX(I,J) +
     1                   WIDTH*WEIGHT*DETJ*A2(I,J)

 201              END DO
 200           END DO

 C             Right hand side
 C             Transformation

              CALL KASET1(STR_GLOB, MCRD)
              DO 202 I=1, MCRD
                 DO 203 J=1, MCRD

                    STR_GLOB(I) = STR_GLOB(I) + THETA(I,J)*STRESS(J)

 203              END DO
 202           END DO


              DO 230 I=1, NDOFEL
                 DO 240 K=1,MCRD

                       RHS(I,1) = RHS(I,1) +
     1                      DETJ*WIDTH*WEIGHT*BT(I,K)*STR_GLOB(K)

 240              END DO
 230           END DO


              IF (NRHS.EQ.2) THEN
                 WRITE(7,*) 'Riks solution not supported by element'
                 CALL FLUSH_(7)
                 CALL XIT
              END IF


              IF (LFLAGS(4).EQ.1) THEN
 C            PERTURBATION STEP
                WRITE(7,*) 'Perturbation step not supported by element'
                CALL FLUSH_(7)
```

```
                CALL XIT
              END IF

C             SAVE OPENING AND STRESSES AT INTEGRATION POINT AS STATE VARIABLES
              SVARS(IINTP+NINTP)   = DU_LOC(1)
              SVARS(IINTP+2*NINTP) = DU_LOC(2)
              SVARS(IINTP+3*NINTP) = STRESS(1)
              SVARS(IINTP+4*NINTP) = STRESS(2)

 100        END DO

         ELSE
            WRITE(7,*) 'Only static procedure supported by element'
            CALL FLUSH_(7)
            CALL XIT
         END IF

      ELSE IF (LFLAGS(3).EQ.4) THEN

         DO I=1, NDOFEL

               AMATRX(I,I)= 1.0d0

         END DO

      ELSE

         WRITE(7,*) 'Only normal incrementation supported by element'
         CALL FLUSH_(7)
         CALL XIT

      END IF
      RETURN
      END


C-----------------------------------------------------------------------------
      subroutine ktracn(RDISP, PROPS, STRESS, DDSDDR, MCRD, SVARS,
     1     NSVARS, IINTP, NINTP, KINC, JELEM)

      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.D0, TWO=2.0D0, ONE= 1.0D0, THREE= 3.0d0)

      DIMENSION PROPS(*), RDISP(MCRD), STRESS(MCRD), DDSDDR(MCRD, MCRD)
      DIMENSION SVARS(NSVARS)

      data ifirst/0/
      data iopen/0/
      data iclose/0/
      save ifirst, nodefirst, iopen, iclose

c     REAL INPUT PROPERTIES
      dJss =    props(1) !Increase in fracture toughness
      deltac =  props(2) !Max crack bridging opening
      delta1 =  props(3) !Initial linear decrease/increase, softening afterwards
      dJ0    =  props(4) !Value of J0 (from measurements)
      fac1   =  props(5) !Stress increase factor for power law
      penalty = props(6) !Penalty factor on contact

      sigma0 = 1.5d0*dJss/delta1*sqrt(delta1/deltac)
      slope1 = -dJss/(delta1*deltac*sqrt(delta1/deltac))

      fac = dJss/(two*sqrt(deltac))
      sigma1 = fac/sqrt(delta1)
      slope = sigma1/delta1

c     J0 is included separately now
c     with zero start power law
```

```
      alpha = 100.0
      delta2 = dJ0/(sigma0*fac1)*(alpha+1)/alpha

c     Code checks for change in opening status. If too many contact points change status,
c     increment can be restarted
c     Checks for opening/closing behaviour
C     get values from state variables

      iold = SVARS(IINTP)
      stressold = SVARS(IINTP + 4*NINTP)
      rdispold  = SVARS(IINTP + 2*NINTP)

c     Code checks for change in contact status. If too many contact points change status,
c     increment can be restarted
c     Checks for opening/closing behaviour

      IF (ifirst.eq.0) THEN
         ifirst = 1
         NODEFIRST = NODE
c         write(7,*) dJss, deltac, delta1, dJ0, fac1, penalty
c         write(7,*) sigma0, sigma1, slope, slope1, delta2, fac
      END IF

c     new increment detection (includes restart) to count contact changes
      IF (NODE.EQ.NODEFIRST.AND.KIT.EQ.1) THEN
         iopen = 0
         iclose = 0
      END IF


c     Check for increasing opening displacement (start in increment 2,
c     once all contact points are closed)
c     Only applies in opening stage (rdisp < 0)
c     Not included right now (KINC.GE.1000)

      IF (rdisp(2).LT.rdispold.AND.rdisp(2).GT.delta2.AND.
     1     KINC.GE.1000) THEN

c     Elastic unloading and reloading
         stress(2) = stressold/rdispold*rdisp(2)
         ddsddr(2,2) = stressold/rdispold
         write(7,*) 'Elastic unloading encountered'

      ELSE
c     Check for penetration of surfaces and indicate status
         if (rdisp(2).LT.zero) then
c
c           write(7,*) 'Area I'
            stress(2) = penalty*slope*rdisp(2)
            ddsddr(2,2) = penalty*slope
            lOpenClose = 0
c
c     Check for opening of crack
c     Stresses will be negative (tension)

c     First slope bit (different from square root law)
         else  if (rdisp(2).GE.zero.and.rdisp(2).LT.delta2) then

c           Initial increase
c               write(7,*) 'Area II'
               stress(2) = fac1*sigma0*
     1                           (1.0-((delta2-rdisp(2))/delta2)**alpha)
               ddsddr(2,2) = fac1*alpha*sigma0/delta2*
     1                           (((delta2-rdisp(2))/delta2)**(alpha-1))


               lOpenClose = 1
```

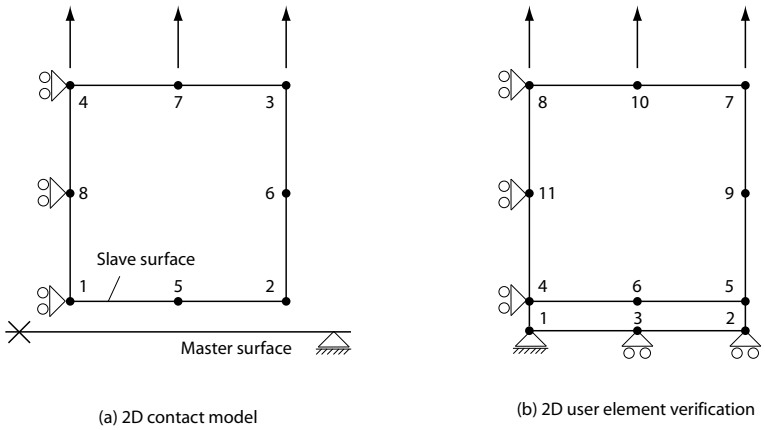
```
c     Softening behaviour

         else if (rdisp(2).GE.delta2.and.
     1                   rdisp(2).LT.(delta1+delta2)) then

c            write(7,*) 'Area III'
               stress(2) = sigma0 + slope1*(rdisp(2)-delta2)
               ddsddr(2,2) = slope1
               lOpenClose = 2


         else if (rdisp(2).GE.(delta1+delta2).
     1                   and.rdisp(2).LT.(deltac+delta2)) then

c            write(7,*) 'Area IV'
               stress(2) = fac/sqrt((rdisp(2)-delta2))
               ddsddr(2,2) = -fac/2*((rdisp(2)-delta2))**(-3.d0/2.d0)
               lOpenClose = 3

         else if (rdisp(2).GT.(deltac+delta2)) then

c            write(7,*) 'Area V'
               stress(2) = 0
               ddsddr(2,2) = 0
               lOpenClose = 4
c
         end if


       END IF

       IF (lOpenClose.NE.iold.AND.iold.EQ.0.AND.KINC.GT.3) THEN

c     Restart if more than one contact pair opens (iold=0) in 2nd call
c          IF (KIT.EQ.2) THEN
             iopen = iopen + 1
c          END IF

         write(7,*) 'Status: iopen=', iopen, 'at int point', IINTP,
     1           'in element', JELEM, 'and increment', KINC
         write(7,*) 'lOpenClose =',lOpenclose,'iold=',iold,'KIT=', KIT

c     Possible restart procedure
c          IF (iopen.gt.1) THEN
c        write(7,*) 'Too many contact openings: reduce increment'
c          END IF

       END IF

c     Restart if one contact pair closes (iold=1/lOpenClose=0)
       IF (lOpenClose.EQ.0.AND.iold.EQ.1.AND.KINC.GE.2) THEN

         iclose = iclose + 1
         write(7,*) 'Status: iclose=', iclose, 'at int point', IINTP,
     1         'in element', JELEM, 'and increment', KINC
         write(7,*) 'lOpenClose =',lOpenclose,'iold=',iold,'KIT=', KIT

         IF (iclose.gt.0) THEN
                write(7,*) 'Elastic unloading possible: reduce increment'
         END IF

       END IF

c     Restart with PNEWDT (if PNEWDT less than 1)
       IF (iclose.gt.0) THEN
                PNEWDT  = 1.0
       ELSE IF (iopen.gt.4) THEN
         PNEWDT = 1.0
```

```
      END IF

c     Sign definition (bridging stress acts as closure stress on structure
c     as in contact analysis)
c     Stiffness matrix according to ABAQUS definition: -dF/du!!

      stress(2) = - stress(2)


C     State variable update
      SVARS(IINTP) = lOpenClose

      return
      end

C-------------------------------------------------------------------------------------------
      subroutine KASET1(DMATRIX, IDIMX)

      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0)

      DIMENSION DMATRIX(IDIMX)


      DO i=1, IDIMX

            DMATRIX(i) = ZERO

      END DO

      RETURN
      END


C-------------------------------------------------------------------------------------------
      subroutine KASET2(DMATRIX, IDIMX, IDIMY)

      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0)

      DIMENSION DMATRIX(IDIMX, IDIMY)

      DO I = 1, IDIMX
         DO J = 1, IDIMY

               DMATRIX(I,J) = ZERO

         END DO
      END DO

      RETURN
      END
C-------------------------------------------------------------------------------------------
```

E Cohesive element verification

E.1 Verification of the 2D element

Figure 24 shows the simple two element model comparing the previously written subroutine UINTER [14] and the new user element. Both models give identical results. The nodes 1-6 for the user element initially coincide in their position: here they are plotted apart for a better understanding of the node numbering.



(a) 2D contact model

(b) 2D user element verification

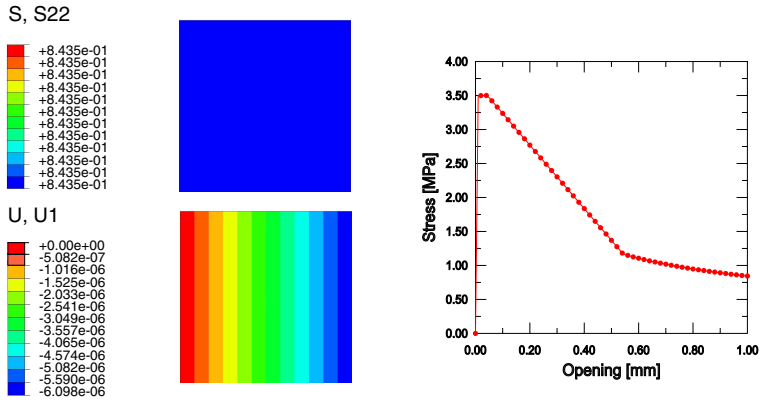
Figure 24. 2D user element model

The input deck is simple and given in the following for the two-element problem. Note that the element is under plane stress condition (element, type=CPS8) to give comparable results to the 3D model with its boundary conditions.

```
*HEADING
*NODE
1, 0.0, 0.0
2, 1.0, 0.0
3, 0.5, 0.0
4, 0.0, 0.0
5, 1.0, 0.0
6, 0.5, 0.0
7, 1.0, 1.0
8, 0.0, 1.0
9, 1.0, 0.5
10, 0.5, 1.0
11, 0.0, 0.5
*NSET, NSET=ALL
8, 10, 9
*user element, type=U6, nodes=6, coordinates=2, i properties=2, properties=7, variables=15
1, 2
*ELEMENT, TYPE=U6, ELSET=ALL
1, 1, 2, 3, 4, 5, 6
*ELEMENT, TYPE=CPS8, ELSET=eall
2, 4, 5, 7, 8, 6, 9, 10, 11
*SOLID SECTION, ELSET=EALL, MATERIAL=MAT1, ORIENTATION=ORIENT1
1.0
*UEL PROPERTY, ELSET=ALL
1.65, 1.0, 0.5, 0.15, 1.0, 1000.0, 1.0, 3
1
*MATERIAL, name=mat1
*ELASTIC, TYPE=ENGINEERING CONSTANTS
41.5E3, 9.5E3, 9.5E3, 0.3, 0.3, 0.3, 15.8E3, 3.65E3
3.65E3
```

```
*orientation, name=orient1
1.0, 0.0, 0.0, 0.0, 1.0, 0.0
1, 0.
*BOUNDARY
1, 1, 2
2, 2, 2
3, 2, 2
4, 1, 1
11, 1, 1
8, 1, 1
*STEP, INC=100, NLGEOM
*STATIC
0.01, 1.0,,0.01
*MONITOR, DOF=2, NODE=4
*CONTROLS, PARAMETERS=TIME INCREMENTATION
7, 10, 9, 16, 10, 4, 20, 10, 6
*BOUNDARY
8, 2, 2, 1.0
7, 2, 2, 1.0
10, 2, 2, 1.0
*OUTPUT, FIELD, FREQ=1
*ELEMENT OUTPUT
S,
E,
*NODE OUTPUT
U,
RF,
*OUTPUT, HISTORY, FREQ=1
*NODE OUTPUT, NSET=ALL
U2,
RF2
*end step
```

Figure 25 shows the results for the test case as created by ABAQUS/CAE. As the stiffness of the top element is large compared to the bridging stress input, the stress distribution in loading direction appears constant. The displacement field in x-direction, which is caused by the Poisson's effect, varies linearly. For the simple case of uniform deformation (the same displacement boundary condition on all top nodes), the original bridging law is obtained from the reaction forces, element area and the opening of the element.



(a) Contour plots (b) History plot

Figure 25. Verification of 2D user element

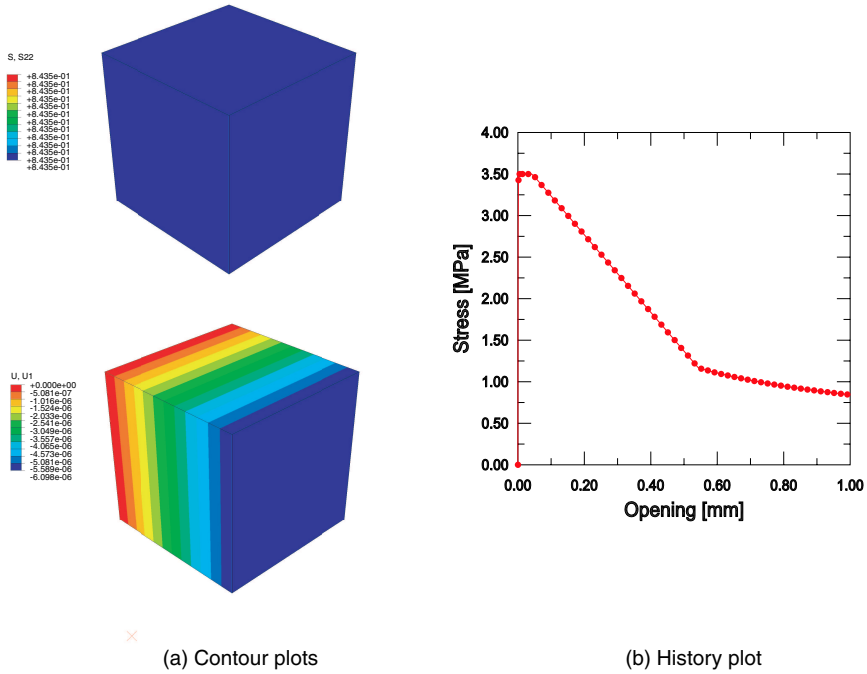
E.2 Verification of the 3D element

The input deck is simple and given in the following for the two element test analogous to the 2D model:

```
*HEADING
*NODE
1, 0.0, 0.0, 1.0
2, 1.0, 0.0, 1.0
3, 1.0, 0.0, 0.0
4, 0.0, 0.0, 0.0
5, 0.5, 0.0, 1.0
6, 1.0, 0.0, 0.5
7, 0.5, 0.0, 0.0
8, 0.0, 0.0, 0.5
9, 0.0, 0.0, 1.0
10, 1.0, 0.0, 1.0
11, 1.0, 0.0, 0.0
12, 0.0, 0.0, 0.0
13, 0.5, 0.0, 1.0
14, 1.0, 0.0, 0.5
15, 0.5, 0.0, 0.0
16, 0.0, 0.0, 0.5
17, 0.0, 1.0, 1.0
18, 1.0, 1.0, 1.0
19, 1.0, 1.0, 0.0
20, 0.0, 1.0, 0.0
21, 0.5, 1.0, 1.0
22, 1.0, 1.0, 0.5
23, 0.5, 1.0, 0.0
24, 0.0, 1.0, 0.5
25, 0.0, 0.5, 1.0
26, 1.0, 0.5, 1.0
27, 1.0, 0.5, 0.0
28, 0.0, 0.5, 0.0
*NSET, NSET=ALL
8, 10, 9
*NSET, NSET=BOTTOM
1, 2, 3, 4, 5, 6, 7, 8
*NSET, NSET=TOP
17, 18, 19, 20, 21, 22, 23, 24
*NSET, NSET=SIDEX
1, 4, 8, 9, 12, 16, 17, 20, 24, 25, 28
*NSET, NSET=SIDEZ
1, 2, 5, 9, 10, 13, 17, 18, 21, 25, 26
*user element, type=U16, nodes=16, coordinates=3, i properties=2, properties=7, variables=63
1, 2, 3
*ELEMENT, TYPE=U16, ELSET=ALL
1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16
*ELEMENT, TYPE=C3D20, ELSET=eall
2, 9, 10, 11, 12, 17, 18, 19, 20, 13, 14, 15, 16, 21, 22, 23,
24, 25, 26, 27, 28
*SOLID SECTION, ELSET=EALL, MATERIAL=MAT1, ORIENTATION=ORIENT1
*UEL PROPERTY, ELSET=ALL
1.65, 1.0, 0.5, 0.15, 1.0, 1000.0, 1.0, 3
1
*MATERIAL, name=mat1
*ELASTIC, TYPE=ENGINEERING CONSTANTS
41.5E3, 9.5E3, 9.5E3, 0.3, 0.3, 0.3, 15.8E3, 3.65E3
3.65E3
*orientation, name=orient1
1.0, 0.0, 0.0, 0.0, 1.0, 0.0
1, 0.
*BOUNDARY
BOTTOM, 2, , 0.0
SIDEX, 1, , 0.0
SIDEZ, 3, , 0.0
*STEP, INC=100, NLGEOM
```

```
*STATIC
0.01, 1.0,
*MONITOR, DOF=2, NODE=9
*CONTROLS, PARAMETERS=TIME INCREMENTATION
7, 10, 9, 16, 10, 4, 20, 10, 6
*BOUNDARY
TOP, 2, 2, 1.0
*OUTPUT, FIELD, FREQ=1
*ELEMENT OUTPUT
S,
E,
*NODE OUTPUT
U,
RF,
*OUTPUT, HISTORY, FREQ=1
*NODE OUTPUT, NSET=ALL
U2,
RF2
*end step
```

The stress and displacement field are identical to the results presented for the 2D model in the previous appendix. For the simple case of uniform deformation (the same displacement boundary condition on all top nodes), the original bridging law is obtained by summing up all reaction forces on the top of the element and recording the opening of the element.



(a) Contour plots (b) History plot

Figure 26. Verification of 3D user element

Mission

To promote an innovative and environmentally sustainable technological development within the areas of energy, industrial technology and bioproduction through research, innovation and advisory services.

Vision

Risø's research **shall extend the boundaries** for the understanding of nature's processes and interactions right down to the molecular nanoscale.

The results obtained shall **set new trends** for the development of sustainable technologies within the fields of energy, industrial technology and biotechnology.

The efforts made **shall benefit** Danish society and lead to the development of new multi-billion industries.