Spring 2015

Comprehensively simulating the mixed-mode progressive delamination in composite laminates

Zhenyuan Gao
Purdue University

Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Zhenyuan Gao

Entitled

Comprehensively Simulating the Mixed-Mode Progressive Delamination in Composite Laminates

For the degree of Master of Science in Aeronautics and Astronautics

Is approved by the final examining committee:

Wenbin Yu
Chair

Vikas Tomar

Arun Prakash

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Wenbin Yu

Approved by: Wayne Chen 4/27/2015

Head of the Departmental Graduate Program Date

COMPREHENSIVELY SIMULATING THE MIXED-MODE PROGRESSIVE DELAMINATION IN COMPOSITE LAMINATES

A Thesis

Submitted to the Faculty

of

Purdue University

by

Zhenyuan Gao

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aeronautics and Astronautics

May 2015

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Wenbin Yu for all the helpful discussions and advice he provided during my thesis writing. I'd like to thank Liang Zhang for the help in revising this thesis and providing advices in scientific writing. I'd like to thank Dr. Vikas Tomar and Dr. Arun Prakash for their effort as my committee members, and all my colleagues in Dr. Yu's group for their support during my preparation for graduation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

SYMBOLS

a_0	crack length
a_1, a_2	power law variables
b	width of specimen
c	lever length
c_d	dilatational wave speed
d	damage parameter
h	half thickness of specimen
l	half length of specimen
n	power law parameter
r	separation ratio
t	thickness of cohesive layer
$\boldsymbol{D}$	stiffness matrix
F	loading function
$G_{\mathrm{Ic}}, G_{\mathrm{IIc}}$	Mode I, Mode II critical fracture energy release rate
K_p	penalty stiffness
K_{pm}	effective penalty stiffness
L_c	characteristic length of element
N_i	shape function
α	maximum separation reached in loading history
β	displacement mode-mixing ratio
η	exponent of the B-K criterion
σ_0	normal direction strength of layer
σ_m^0	effective normal strength
τ_0	shear direction strength of layer
τ_m^0	effective shear strength

δ_0	displacement for start of softening
δ_f	displacement for total decohesion
δ_m	effective displacement
$\bar{\bar{\delta}}_{ij}$	Kronecker delta
κ	energy mode mixing ratio
Δ_i	global relative displacement

ABSTRACT

Gao, Zhenyuan MSAA, Purdue University, May 2015. Comprehensively Simulating the Mixed-Mode Progressive Delamination in Composite Laminates. Major Professor: Wenbin Yu.

Delamination, or interlaminar debonding, is a commonly observed failure mechanism in composite laminates. It is of great significance to comprehensively simulate the mixed-mode progressive delamination in composite structures because by doing this, people can save a lot of effort in evaluating the safe load which a composite structure can endure.

The objective of this thesis is to develop a numerical approach to simulating double-cantilever beam (DCB) and mixed-mode bending (MMB) tests and also of specifying/validating various cohesive models. A finite element framework, which consists of properly selecting time integration scheme (explicit dynamic), viscosity, load rate and mass scaling, is developed to yield converged and accurate results.

Two illustrative cohesive laws (linear and power-law) are programmed with a user-defined material subroutine for ABAQUS/Explicit, VUMAT, and implemented into the finite element framework. Parameters defined in cohesive laws are studied to evaluate their effects on the predicted load-displacement curves.

The finite element model, together with the predetermined model parameters, is found to be capable of producing converged and accurate results. The finite element framework, embedded with the illustrative cohesive laws, is found to be capable of handling various interfacial models.

The present approach is concluded to be useful in simulating delamination with more sophisticated material models. Together with the method for determining model parameters, it can be used by computer codes other than ABAQUS.

1. Introduction

1.1 Motivation

Delamination, or interlaminar debonding, is a commonly observed failure mechanism in heterogeneous materials, especially in composite laminates. When a composite laminate is subject to a certain extent of transverse load, either during its manufacturing or in use, progressive delamination may occur within it. Such phenomenon should be avoided because it may harm the performance of a composite structure such as its load-carrying capability and structural integrity. One major challenge is to predict the onset of such phenomenon: first, it is often difficult or expensive to experimentally quantify delamination, especially with mixed-mode delamination; second, the interfacial properties significantly affect delamination and vary case by case. All these lead one to seek for a comprehensive method for numerically simulating the progressive mixed-mode delamination in composite laminates.

1.2 Literature Review

Several experimental methods have been designed to measure the progressive delamination in composite laminates, either pure mode or mixed-mode. The double-cantilever beam (DCB) and the end-notched flexure (ENF) tests were first designed especially for pure mode I and II delamination, respectively (see Figure 1.1). The mixed-mode bending (MMB) test [1] was later designed to quantify the mixed-mode delamination (Figure 1.2). These three testing methods require the same specimen (i.e., a pre-cracked composite beam consisting of two plies glued with one layer of adhesive) but different loading conditions:

1. in the DCB test, two opposite vertical forces opening the crack are applied at the free end of a cantilever beam;

2. in the ENF test, a downward force is applied at the middle of a simply supported beam;

3. in the MMB test, two opposite vertical forces are applied at the cracked end and the middle of a simply supported beam, respectively, with a loading lever placed above the beam, and the length of the loading lever is changed for each test to yield designated mode-mixing ratio by adjusting the ratios between the two forces.

During a test, the load-displacement curve at a load point is recorded, from which the critical fracture energy release rate during progressive delamination can be extracted.

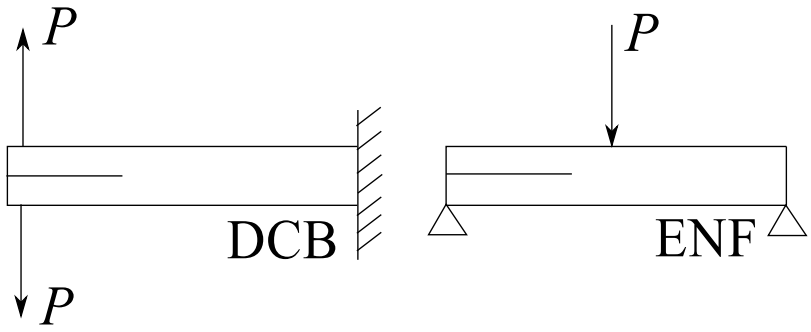

Figure 1.1. Schematics of the DCB and ENF tests

Several direct and indirect numerical approaches have been developed to model the delamination in composite laminates. A direct approach involves directly computing the strain energy release rate with the theory of fracture mechanics, from some finite element analysis (FEA) results. The virtual crack closure technique (VCCT) is a widely used direct approach. It involves making use of the crack-closure integral, or to say, the total work needed to close a crack. Rybicki and Kanninen [2] first proposed the VCCT for 2D problems. Shivakumar et al. [3] enabled this approach to handle

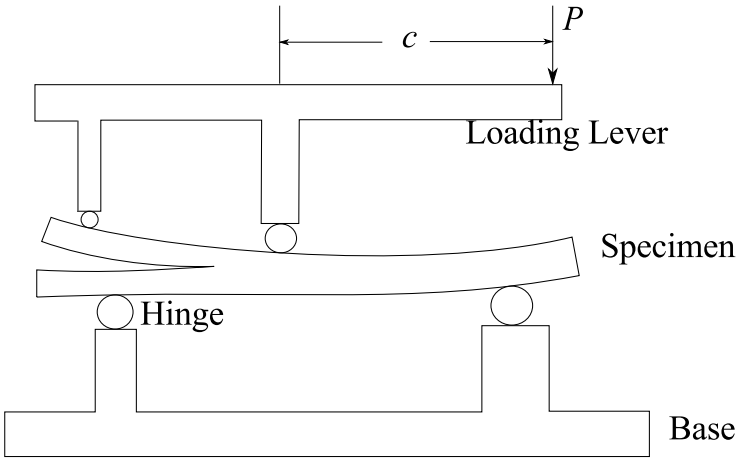

Figure 1.2. The mixed-mode bending test

3D problems. It can predict the onset and stability of delamination but is incapable of predicting the damage initiation because it requires initial delamination defined a priori. Meanwhile, VCCT implicitly assumes that the crack growth is self-similar, which means the crack front remains straight throughout the delamination process. This, however, is often not the case because even in DCB tests, the crack surfaces will become curved after crack propagation [4]. In contrast, the indirect approaches, which make use of some idealization of the interface (e.g., a cohesive zone model), can overcome these drawbacks.

Numerous researchers have developed various cohesive zone models which can be implemented in various finite element codes. A cohesive zone model consists of a cohesive element and a cohesive law. A cohesive element acts as a geometric representation of the interface and is often placed between two glued plies, where delamination is expected to occur. Cohesive elements can be classified into continuum and point cohesive elements. A point cohesive element [5] acts as a spring connecting two crack surfaces. It can be placed anywhere delamination may occur, but the size of other elements used in the model has to be very small to yield accurate results. A set of continuum cohesive elements can be modelled by the usual elements provided in existing finite element packages, and form into a continuous adhesive layer (see

Figure 1.3), where in the thickness direction there is only one element. A cohesive law specifies the constitutive behavior of a cohesive element as a function of the tractions and the relative displacements. A typical pure mode traction-relative displacement curve consists of a delamination initiation and a softening (or propagation) part. The area underneath the traction-relative displacement curve has to equal the interfacial critical fracture energy release rate such that the cohesive zone model and the theory of fracture mechanics are energetically equivalent. Glennie [6] first introduced a strain-rate dependent cohesive zone model. It specifies the tractions acting on an interface as a function of the crack opening rate. Several authors later developed various other cohesive laws such as linear softening, progressive softening, and regressive softening ones (see Ref. [7] for more details). Despite differences, all these laws have the aforementioned the delamination initiation and softening (or propagation) parts in correspondence to the two stages of the delamination evolution.

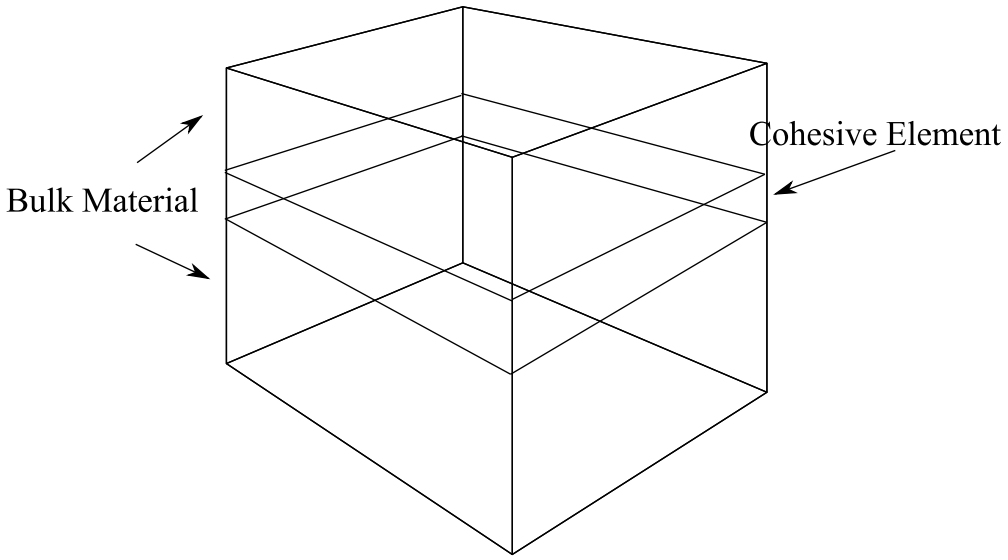

Figure 1.3. Cohesive element

Several researchers have established different finite element models to simulate the progressive delamination in composite laminates, with different simplifications. Camanho et al. [8] idealize the glued two plies as two shells and meshed them with

shell elements, and Turon et al. [9] identify the problem as a plane stress one and meshed the plies with 2D plane stress elements. Although the predictions by both models agree well with the experimental results, the simplifications involved in them are actually improper. In the former case, although the thickness-to-width ratio of the beam is very small, the structure still cannot be idealized as a shell because the effect of the beam thickness on the deformation of cohesive elements cannot be neglected. In the latter case, the problem is a plane stress one only if the thickness-to-width ratio of the beam is very large. This, however, is contrary to the experimental setup in Ref. [1]. Meanwhile, the problem is not a plane strain one either because the deformation in the beam width direction is not constrained. Therefore, it is necessary to model the beam as it is, i.e., a 3D body, and to mesh the 3D body with 3D brick elements. In this case, some issues may arise and await resolution:

1. an implicit static integration scheme may not give converged results, and one has to use an explicit dynamic one;

2. once an explicit dynamic integration scheme is adopted, the laminate density and the load rate have to be properly selected to yield converged and accurate results;

3. some viscosity terms and extremely fine meshes around the interface need to be introduced to solve the problems of non-uniqueness of solution and snap-back instability in the presence of high interface strength and mixed-mode delamination [10].

1.3 Objectives

The objective of this thesis is to develop a comprehensive numerical method for simulating DCB and MMB tests and also for specifying and validating various cohesive models. The finite element model is established following the experimental setup in Ref. [1]. The load rate and the extent of mass scaling are properly selected to yield

converged results and to approximate a quasi-static loading process. Some viscosity terms and fine meshes around the interface are introduced to avoid non-uniqueness of solution and snap-back instability in the presence of high interface strength and mixed-mode delamination. Some fundamentals of the cohesive zone model are briefly introduced. A user-defined material subroutine for ABAQUS/Explicit, VUMAT, is used to program a cohesive law into the finite element model. A linear and a power law cohesive law are chosen as illustrative examples. The predictions by the present approach are compared with the experiment results for validation purposes. The effects of the interfacial properties on the predicted load-deflection curves are evaluated. Although the present approach is developed with ABAQUS/Explicit, it is also applicable to other finite element codes.

2. Finite Element Framework

In this chapter, a finite element framework will be established with its finite element model built based on the experimental setup of MMB and DCB tests in Ref. [8]. The method of properly selecting the time integration scheme, viscosity and mass-scaling factor will be presented such that the finite element framework, together with a cohesive zone model, can produce converged and accurate results.

2.1 Experimental Setups

The experiment is mainly based on the MMB test apparatus. This is because one can convert a MMB test to a DCB test by removing the loading lever in a MMB test, and by applying loads at the cracked end of the specimen. MMB tests use virtually the same specimen for all mode-mixing ratios, making it unnecessary to adjust the specimen throughout all the simulations.

In a MMB test, the load is not directly applied on the specimen but on the loading arm above it. The experiment results on such tests show that the relation between the load acting on the arm and the displacement at the loading point. For different mode-mixities, different initial crack lengths are used, which is obtained from experiments. The lever length is adjustable and can be calculated using the mode-mixing ratio and specimen length. The weight of the lever is negligible.

Different mode-mixing ratios can be achieved simply by changing the length c of the loading arm. The value of c can be obtained analytically [8]. First, simple beam theory analysis of a DCB specimen leads to:

$$G_{\mathrm{I}} = \frac{12a_0^2P_{\mathrm{I}}^2}{b^2h^3E_{11}}, \qquad G_{\mathrm{II}} = \frac{9a_0^2P_{\mathrm{II}}^2}{16b^2h^3E_{11}} \tag{2.1}$$

where G_{I} and G_{II} are the Modes I and II energy release rates, respectively, P_{I} and P_{II} are the forces acting at the cracked end and the middle of the specimen, respectively, and a_0, b, and h are the initial crack length, width of specimen, and half thickness of specimen, respectively. By solving the equilibrium equation for a beam subject to three loads P, P_{I} and P_{II} and also by spliting mixed-mode loading into Mode I and Mode II loading, P_{I} and P_{II} can be expressed related to P by:

$$P_{\mathrm{I}} = \left(\frac{3c-l}{4l}\right)P, \qquad P_{\mathrm{II}} = \left(\frac{c+l}{l}\right)P \tag{2.2}$$

Substituting Eq. (2.2) into Eq. (2.1) yields:

$$G_{\mathrm{I}} = \frac{3a_0^2P^2}{4b^2h^3l^2E_{11}}\left(3c-l\right)^2, \qquad G_{\mathrm{II}} = \frac{9a_0^2P^2}{16b^2h^3l^2E_{11}}\left(c+l\right)^2 \tag{2.3}$$

Dividing G_{I} by G_{II} in Eq. (2.3) gives:

$$\frac{G_{\mathrm{I}}}{G_{\mathrm{II}}} = \frac{4}{3}\left(\frac{3c-l}{c+l}\right)^2 \tag{2.4}$$

The expression for the length of lever c can then be obtained as a function of mode-mixing ratio and length of specimen:

$$c = l\frac{\frac{1}{2}\sqrt{3\left(\frac{1-\kappa}{\kappa}+1\right)}}{3-\frac{1}{2}\sqrt{3\left(\frac{1-\kappa}{\kappa}\right)}} \tag{2.5}$$

where:

$$\kappa = \frac{G_{\mathrm{II}}}{G_{\mathrm{I}}+G_{\mathrm{II}}} \tag{2.6}$$

is the mode-mixing ratio in terms of energy release rate. Eq. (2.5) gives the values of c for the corresponding mode-mixities (Table 2.1). Table 2.2 lists the specimen geometry adopted by Reeder and Crews [1].

Table 2.1. Lever lengths for different mode-mixities

κ	20%	50%	80%
c(mm)	109.4	44.4	28.4

Table 2.2. Specimen geometry

	length ($2l$)	width (d)	thickness ($2h$)	crack length (a_0)
dimension (mm)	102	25.4	3.12	33

The DCB test shares the same finite element model with the MMB test, except that the loading lever is removed and that the load acts at the cracked end of the specimen with different boundary conditions.

2.2 Finite Element Model

The MMB test specimen is a plate-like 2-ply laminate. All elements except for the ones in the cohesive layer will be 8-node brick elements. Cohesive elements are incorporated using user subroutine VUMAT. To get a specimen model that can lead to converged and accurate results, some model parameters need to be adjusted and tested using the DCB test.

By modeling the arm as a rigid body, and by tying the two arm braches to the specimen and prescribing a displacement at the end of the arm varying from 0 to desired value, such test can be simulated with ABAQUS (Figure 2.1). The specimen is pinned in all three directions at the cracked end, and pinned in two directions at the other end, allowing it to move in the longitudinal direction.

Cohesive elements in the finite element model are bonded to the brick elements in the bulk material. Their sizes in shear directions have to be the same as the corresponding brick elements to be easily bonded. The thickness of a cohesive element

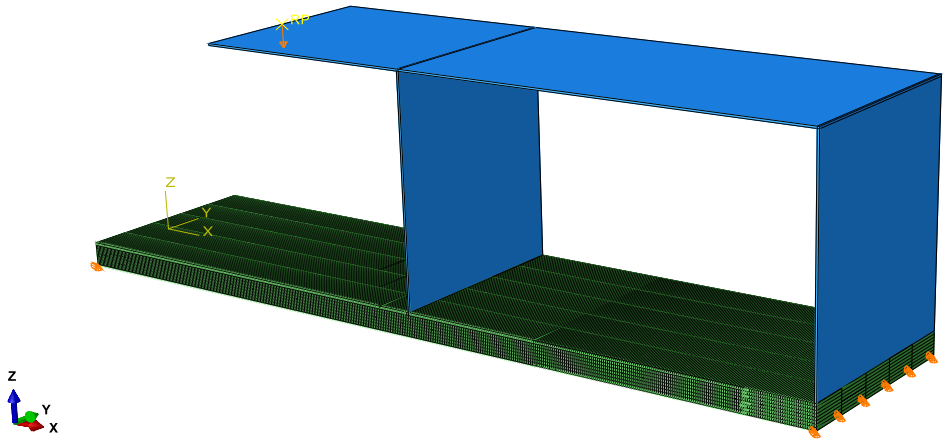

Figure 2.1. Finite element model for the MMB test

is often set to 0 or a very small value because it is merely a fictitious layer. Since cohesive element is actually a 8-node element, the so-called constitutive thickness in ABAQUS can take nonzero values. This constitutive thickness will be used in constitutive relations within a cohesive element as the initial separation in the thickness direction. It is also the characteristic element length of a cohesive element which affects the stable time increment of the finite element analysis. This is because in ABAQUS, the stable time increment is approximated as [11]:

$$\Delta t = \frac{L_c}{c_d} \tag{2.7}$$

Thus when modeling the MMB test specimen, a zero-thickness cohesive layer is inserted between the upper and the lower layers where the pre-crack is present. At the same time, the constitutive thickness of this cohesive layer can be specified. The simulation results will not be affected much by the constitutive thickness value as long as the value is very small compared with the specimen thickness, while the computation time will be affected due to the change in the characteristic element length.

2.3 Time Integration Scheme and Viscosity

Simulations using cohesive elements often experience convergence difficulties after the crack starts propagating. Specially, in implicit finite element analysis, the computation will possibly fail to converge when the traction in a cohesive element reaches the critical value. The instability occurring when the traction attains its maximum in a cohesive element is the main reason for difficulty of convergence. By using dynamic steps in ABAQUS/Explicit, one can avoid most of these convergence problems because it simulates a real process of delamination process. The displacement at the load point can be easily prescribed to yield results for loading magnitude. As long as the dynamic steps are quasi-static, the results of the explicit method can agree well with experiment results.

For the explicit method, solution at the crack propagating point might also be very unrealistic, which always deviate from the other points on the traction versus separation curve. By adjusting other parameters of the finite element model, convergence problem can be overcome.

In addition, some numerical approaches have been introduced to solve or alleviate this problem. By first prescribing a magnitude of opening at the crack propagating point, one can avoid the instability at the critical point. Meanwhile, incremental approaches like Riks method can be applied to provide better convergence quality [12].

Another effective way is to include a small viscosity term in the cohesive law, which is simple and productive. By adjusting the value of viscosity, one can also get better curve quality when modeling with 3D elements, without affecting the results too much.

By adopting ABAQUS/Explicit integration scheme and by applying viscosity to simulation, one can guarantee the convergence of the finite element analysis of delamination of composite laminates.

2.4 Mass Scaling and Load Rate

Due to the dynamic nature of delamination problems, one should ensure that the problem is solved in a quasi-static set-up. This means that the kinetic energy should not take up much ratio in the total energy at most times during the analysis. Otherwise, the results will be unstable and thus inaccurate. This ratio can be adjusted using different load rates and material densities. But applying a real material density may lead to much more computation time. By increasing the density, one can also reduce the solving time. This is known as mass scaling for explicit dynamic analysis.

Mass scaling can affect the computational time because the stable time increment in ABAQUS analysis is approximated as [11]:

$$\Delta t = \frac{L_c}{c_d} \tag{2.8}$$

where L_c is the smallest characteristic element length, and c_d is dilatational wave speed. For linear elastic material:

$$c_d = \sqrt{\frac{E}{\rho}} \tag{2.9}$$

If the density is increased from ρ to $f\rho$, where f is the mass-scaling factor, the wave speed is decreased from c_d to $\frac{c_d}{\sqrt{f}}$. The stable time increment will vary from Δt to $\sqrt{f}\Delta t$, causing the number of increments to be reduced by $\sqrt{f}$ times. Together with the control of load rate, mass scaling can help to analyze the model in shorter time period, and to keep the analysis in quasi-static state.

Since increasing the load rate or the density/mass scaling factor will achieve similar effects, one can first increase the density of the specimen to get a significantly reduced computation time, and then adjust the load rate/step time to get the dynamic energy to total energy ratio small enough. It can also be done in the opposite order. Both approaches are capable of producing economic quasi-static solutions.

By choosing suitable coupling between mass scaling and load rate, one can significantly improve the accuracy of the finite element solution, and saving computation cost at the same time.

2.5 Mesh Density

For complex finite element model, the accuracy of the solution highly depend on the mesh density. Because the energy dissipated during the fracture process is proportional to the volume of failed elements rather than the area of fracture surface. With a fixed displacement for total decohesion (δ_f), the energy dissipated will decrease upon mesh refinement, leading to unreal results as the mesh is fine enough [13].

To alleviate this problem, a way to calculate the strength limit with respect to G_c is proposed as:

$$\delta_f = \frac{2G_c}{\sigma_0 L_c} \tag{2.10}$$

where L_c is a characteristic length of the element, which will mostly be determined by the constitutive thickness of the cohesive layer. Although this will not solve the problem eventually, finer meshes will surely improve the curve quality in the simulation.

It is also worth noticing that using different element sizes in one model will also cause accuracy problems. If the DCB specimen is modeled using fine meshes near crack tip and coarse meshes in other parts, the results will be far from realistic values. Finer meshes can surely help with the quality in terms of load-displacement curve in the simulation of delamination but can also potentially reduce the dissipated energy and thus results of critical load.

3. Cohesive Zone Model

In this chapter, some fundamentals of the cohesive zone model (specifically, cohesive law) will be briefed, and the method of programming these illustrative cohesive laws with a material user subroutine for ABAQUS/Explicit, VUMAT, will be presented. A cohesive zone model consists of cohesive element and cohesive law.

3.1 Cohesive Element

The cohesive element is based on isoparametric hexahedral solid brick element. In this section, this formulation is reviewed. The overall idea is to use continuum method to model discontinuous composite delamination, where the cohesive element has small thickness compared to the width and length. There is also a hypothesis that fracture in such a failure plane formed by the length and width of the element, fracture is caused by normal stress and two shear stresses acting on it, which has been adopted by many authors [14]. So only these three stresses will be accounted for fracture of the element, while the other three stresses (two in-plane normal stresses and the third shear stress) will be addressed only by in-plane deformation of laminate.

Cohesive element has a near-zero thickness and 8 nodes and can be bonded to top and bottom elements which allow their kinematics to be compatible. Using a global coordinate system e_i that spans the Euclidean space, and local coordinate system denoted by s, n, and t, Figure 3.1 shows the stress state of such a cohesive element. Here n denotes the thickness direction, which corresponds to Mode I failure. s, t denote Mode II and Mode III failure respectively.

Another definition is the relative displacement Δ, which is the function of the displacement of points located on top and bottom surface of the element, u_i^+ and u_i^-:

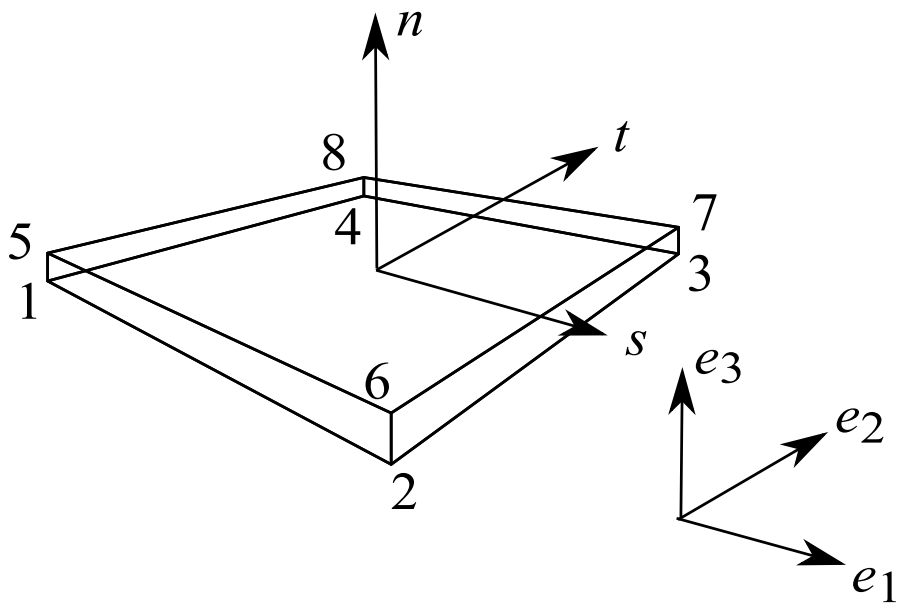

Figure 3.1. 8-node cohesive element

$$\Delta_i = u_i^+ - u_i^- \tag{3.1}$$

And u_i^+ and u_i^- are obtained from:

$$\begin{aligned} u_i^+ &= N_k u_{ki}^+, \qquad k = 5,6,7,8 \\ u_i^- &= N_k u_{ki}^-, \qquad k = 1,2,3,4 \end{aligned} \tag{3.2}$$

where u_{ki}^+ and u_{ki}^- are the displacements of top and bottom nodes in the element in i direction, respectively. N_k are Lagrangian shape functions. Eq. (3.1) can be rewritten as:

$$\Delta_i = \bar{N}_k u_{ki} \tag{3.3}$$

by defining:

$$\bar{N}_k = \begin{cases} N_k, & k = 5,6,7,8 \\ -N_k, & k = 1,2,3,4 \end{cases} \tag{3.4}$$

The tangential plane at a point is spanned by $\boldsymbol{v_\xi}$ and $\boldsymbol{v_\zeta}$, which are results of differentiating global position vectors to local coordinates:

$$v_{\xi_i} = x_{i,s}, \qquad v_{\zeta_i} = x_{i,t} \tag{3.5}$$

where $\boldsymbol{x}$ is obtained for isoparametric elements as:

$$x_i = N_k x_{ki} \tag{3.6}$$

such that the two vectors spanning the tangential plane can be written as:

$$\begin{aligned} v_{\xi_i} &= (N_k x_{ki})_{,\xi} = N_{k,\xi} x_{ki} \\ v_{\zeta_i} &= (N_k x_{ki})_{,\zeta} = N_{k,\zeta} x_{ki} \end{aligned} \tag{3.7}$$

Then the local normal coordinate vector is the vector product of $\mathbf{v}_\xi$ and $\mathbf{v}_\zeta$:

$$\boldsymbol{n} = \frac{\boldsymbol{v_\xi} \times \boldsymbol{v_\zeta}}{\parallel \boldsymbol{v_\xi} \times \boldsymbol{v_\zeta} \parallel} \tag{3.8}$$

The tangential coordinate vectors are:

$$\begin{aligned} \boldsymbol{s} &= \frac{\boldsymbol{v_\xi}}{\parallel \boldsymbol{v_\xi} \parallel} \\ \boldsymbol{t} &= \boldsymbol{n} \times \boldsymbol{s} \end{aligned} \tag{3.9}$$

The element tractions are defined as:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \sigma \end{pmatrix} = \begin{pmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & D_{33} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} \tag{3.10}$$

where $\boldsymbol{D}$ is the constitutive relation matrix, and δ_i is the local relative displacement or element relative displacement with 3 denoting the normal direction.

3.2 Cohesive Law

3.2.1 Overview

A cohesive law specifies the relationship between the tractions and the relative displacements. Delamination process often consists of damage initiation and damage evolution (or softening). One commonly used softening mode is linear softening. Figure 3.2 shows a linear softening cohesive law for one pure mode.

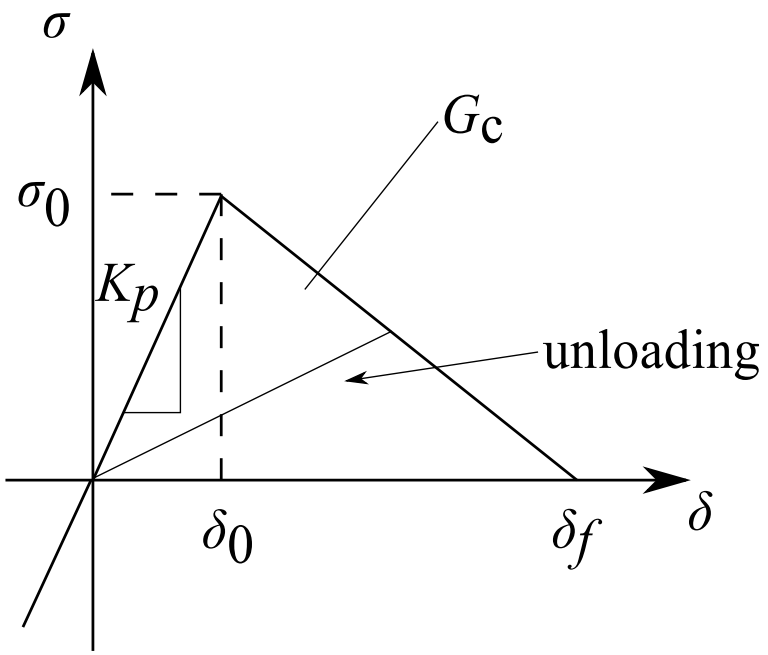

Figure 3.2. Linear cohesive law

Softening models used by others include exponential softening model which has an exponential curve for the whole delamination process, and adaptive softening model which is based on linear softening, but will change its shape as the crack grows. The choice of softening models need to be considered when modeling the test specimen. At the same time, new models can be incorporated with the analysis using user subroutines.

One factor that will affect the accuracy of the results of delamination simulation, as well as the quality of the load-displacement curves, is the elastic property of the

cohesive layer, which is included in many cohesive laws. Before damage initiation, there is usually a relatively linear part of the material property that will hold the two faces of the cohesive elements together. Such property can allow the damage propagation to happen, but not too fast, which resembles the real situation. So if linear softening model, or bi-linear cohesive law, is used, a "penalty stiffness" for this linear part needs to be defined. Such that before damage initiation, this layer will first exhibit elastic property.

Mi et al. [15] suggested that the exact value of displacement for softening onset (δ_0) had negligible effects on the overall response as long as the interface was initially "very stiff". But high penalty stiffness will cause numerical problems. Daudeville et al. [16] suggest the penalty stiffness to be defined as:

$$K_{p1} = \frac{E_3}{t}, \quad K_{p2} = \frac{2G_{13}}{t}, \quad K_{p3} = \frac{2G_{23}}{t} \tag{3.11}$$

where E_3, G_{13}, and G_{23} are the elastic moduli of the composite laminates, and t is the geometric thickness set for the cohesive layer. This comes from an analogy that the cohesive layer is a rich resin zone same as the upper and lower plies just with a smaller geometric thickness. So if the differences of thickness in the two plies and the cohesive layer are not considered, they three should have the same strength overal. This analogy can be used to serve as an approximation of the desired penalty stiffness. So in this work, the penalty stiffness will be around 10^6 N/mm^3 by approximating the geometric thickness of the cohesive layer to be 0.01 mm. That is less than 1% of the thickness of the specimen, so it can give reasonable results. While Dávila et al. [17] reaches a conclusion that, for the specimen used here, a penalty stiffness equal to 10^6 N/mm^3 for all three modes will produce the same results as with higher stiffness and also avoid convergence problems.

3.2.2 Linear Cohesive Law

The behavior within the linear softening model is described as following:

1. elastic loading: for each pure mode, the traction-separation relation will first be linear elastic, and has a penalty stiffness K_p that defines the elastic behavior;

2. softening: after the traction reaches maximum (σ_0 and τ_0), cohesive element enters softening part of the model. Traction will decrease as displacement increases with another linear relation and reach zero at displacement for total decohesion;

3. unloading: when unloading in the softening region, the relation is also linear, whose slope is determined by the maximum displacement reached in loading history (see Figure 3.2);

4. critical fracture energy release rate: the area of the triangle should be equal to the critical fracture energy release rate of that mode, which is from results of experiments;

5. negative displacement: for Mode II and Mode III loading, the negative displacement part of the relation forms an odd function with the positive part. But for Mode I loading, the negative displacement part will still be a linear elastic relation (Figure 3.2). This is to account for the fact that layers can not penetrate each other when compressed.

Therefore, the formulation for linear softening model of each pure mode can be described on the basis of this behavior. First, in the linear elastic zone:

$$D_{ij} = K_p \bar{\delta}_{ij}, \qquad \alpha \leq \delta_0 \tag{3.12}$$

where K_p is the penalty stiffness in this particular pure mode, and α denotes the maximum displacement reached in history. This equation accounts for the linear behavior before damage propagation. When the damage starts to propagate, there is:

$$D_{ij} = \bar{\delta}_{ij} \left[(1-d)K_p + dK_p \frac{\langle -\delta_3 \rangle}{-\delta_3} \bar{\delta}_{i3} \right], \qquad \delta_0 < \alpha < \delta_f \tag{3.13}$$

where $\langle \cdot \rangle$ is the Macauley operator ($\langle x \rangle \equiv (|x|+x)/2$), and d denotes a variable that defines the extent of delamination:

$$d = \frac{\delta_f(\alpha - \delta_0)}{\alpha(\delta_f - \delta_0)} \tag{3.14}$$

and the term $dK_p \frac{\langle -\delta_3 \rangle}{-\delta_3} \bar{\delta}_{i3}$ accounts for the assumption when displacement in the normal direction (δ_3) is negative, the constitutive relation is still linear so that upper and lower faces of the cohesive layer can not penetrate each other. When displacement goes beyond total decohesion, there is:

$$D_{ij} = \bar{\delta}_{i3}\bar{\delta}_{3j}\frac{\langle -\delta_3 \rangle}{-\delta_3}K_p, \qquad \alpha \geq \delta_f \tag{3.15}$$

which states that traction (stress) will be present only when displacement in the normal direction (δ_3) is negative, or there will be no traction because this integration point is completely damaged. Conclusively, the formulation for linear softening model of each pure mode can be described as:

$$D_{ij} = \begin{cases} K_p \bar{\delta}_{ij}, & \alpha \leq \delta_0 \\ \bar{\delta}_{ij}\left[(1-d)K_p + dK_p \frac{\langle -\delta_3 \rangle}{-\delta_3}\bar{\delta}_{i3}\right], & \delta_0 < \alpha < \delta_f \\ \bar{\delta}_{i3}\bar{\delta}_{3j}\frac{\langle -\delta_3 \rangle}{-\delta_3}K_p, & \alpha \geq \delta_f \end{cases} \tag{3.16}$$

With linear softening (or bi-linear cohesive law) described before, each of the pure mode delamination will have the same traction-separation curve in terms of curve shape. The area enclosed in the triangular curve is the pure mode critical fracture energy release rate which can be obtained from experiments. For a loading condition with different delamination modes, a mixed-mode formulation needs to be developed. The mixed-mode formulation developed by Camanho and Dávila [8], as well as some other researchers, features a definition of a scalar variable called effective displacement, δ_m, which is defined as:

$$\delta_m = \sqrt{\delta_1^2 + \delta_2^2 + \langle \delta_3 \rangle^2} = \sqrt{\delta_{\text{shear}}^2 + \langle \delta_3 \rangle^2} \tag{3.17}$$

where δ_i stands for the separation in each of the three modes (the number 3 denotes normal direction of the cohesive layer). With this effective displacement, the delamination behavior at certain traction-separation state can be described as a linear softening model of δ_m.

Since the penalty stiffness is the same for all three modes, the effective linear softening curve will also have the same penalty stiffness. According to different mode-mixing ratios between the two modes, the effective displacement curve will also be different. And the effective strength σ_m^0 will be between σ_0 and τ_0. Thus, a mode-mixing ratio β is defined as:

$$\beta = \frac{\delta_{\text{shear}}}{\delta_3} \tag{3.18}$$

A maximum stress (traction) failure criterion is often enough for this kind of problem, since the layer strengths σ_0 and τ_0 are actually assumed and do not have exact values. Here, to account for the mode-mixing behavior, a quadratic failure criterion is used:

$$\left(\frac{\langle\sigma\rangle}{\sigma_0}\right)^2 + \left(\frac{\tau_1}{\tau_0}\right)^2 + \left(\frac{\tau_2}{\tau_0}\right)^2 = 1 \tag{3.19}$$

The reason both shear directions use the same strength is that, in experiments, Mode III delamination is seldom tested because of the difficulty in applying the load. So the strength in the two shear directions is usually assumed to be the same. This criterion will mainly be used to determine the onset of crack propagation, or rather, be used to determine the displacement of propagation onset δ_m^0.

The penalty stiffness is assumed to be the same for the three directions, so:

$$\delta_3^0 = \frac{\sigma_0}{K_p}, \qquad \delta_1^0 = \delta_2^0 = \frac{\tau_0}{K_p} \tag{3.20}$$

Thus, substituting equations (3.20), (3.18), and (3.17) into Eq. (3.19) gives the expression for this effective displacement of propagation onset:

$$\delta_m^0 = \begin{cases} \delta_3^0 \delta_1^0 \sqrt{\dfrac{1+\beta^2}{(\delta_1^0)^2 + (\beta\delta_3^0)^2}}, & \delta_3 > 0 \\ \delta_{\text{shear}}^0, & \delta_3 \leq 0 \end{cases} \tag{3.21}$$

from which it is clear that when $\beta = 0$, $\delta_m^0 = \delta_3^0$. And when $\beta \to \infty$, $\delta_m^0 = \delta_{\text{shear}}^0$.

Mixed-mode criterion determines at a state other than pure Mode I or pure shear mode, what the relation between pure mode critical fracture energy release rate and the mixed-mode ones will be. Figure 3.3 illustratively shows the mixed-mode criterion.

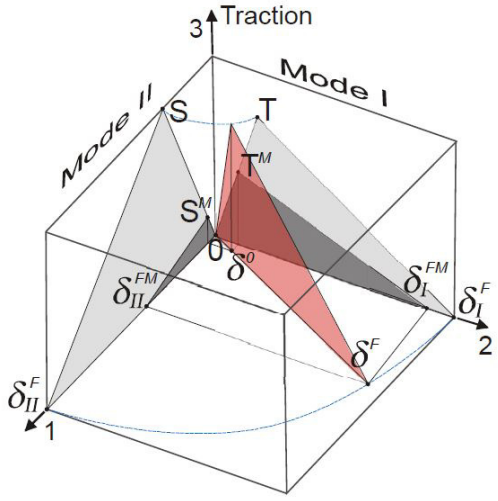

Figure 3.3. Mixed-mode delamination criterion [18]

The mixed-mode criterion proposed by Benzeggagh and Kenane [19] is used to describe the energy relations among different modes. It uses a B-K parameter η obtained from experiments to account for different materials. The criterion is as following:

$$G_{\text{Ic}} + (G_{\text{IIc}} - G_{\text{Ic}}) \left(\frac{G_{\text{shear}}}{G_{\text{T}}} \right)^{\eta} = G_{\text{c}} \tag{3.22}$$

where G_{T} is the total energy release rate:

$$G_{\text{T}} = G_{\text{I}} + G_{\text{shear}} = G_{\text{I}} + G_{\text{II}} + G_{\text{III}} \tag{3.23}$$

Since the critical fracture energy release rate is the area underneath the triangular curve, so:

$$\begin{aligned} G_{\mathrm{I}} &= \frac{K_p \delta_3^f \delta_3^0}{2} \\ G_{\mathrm{shear}} &= \frac{K_p \delta_{\mathrm{shear}}^f \delta_{\mathrm{shear}}^0}{2} \end{aligned} \tag{3.24}$$

The relations between effective displacement and pure mode displacement give:

$$\begin{aligned} \delta_3^0 &= \frac{\delta_m^0}{\sqrt{1+\beta^2}}, & \delta_f^0 &= \frac{\delta_m^f}{\sqrt{1+\beta^2}} \\ \delta_{\mathrm{shear}}^0 &= \frac{\beta \delta_m^0}{\sqrt{1+\beta^2}}, & \delta_{\mathrm{shear}}^f &= \frac{\beta \delta_m^f}{\sqrt{1+\beta^2}} \end{aligned} \tag{3.25}$$

Substituting equations (3.24) and (3.25) into Eq. (3.22) gives the effective displacement for total failure:

$$\delta_m^f = \begin{cases} \dfrac{2}{K_p \delta_m^0} \left[G_{\mathrm{Ic}} + (G_{\mathrm{IIc}} - G_{\mathrm{Ic}}) \left(\dfrac{\beta^2}{1+\beta^2} \right)^{\eta} \right], & \delta_3 > 0 \\ \sqrt{(\delta_1^f)^2 + (\delta_2^f)^2}, & \delta_3 \leq 0 \end{cases} \tag{3.26}$$

d is already defined, the variable for damage evolution in Eq. (3.14). Now with the effective displacement, this definition is changed to:

$$d = \frac{\delta_m^f (\alpha - \delta_m^0)}{\alpha (\delta_m^f - \delta_m^0)} \tag{3.27}$$

The value of d increases from 0 to 1, denoting the extent of damage evolution. Finally, the constitutive relation is:

$$\boldsymbol{\sigma} = \boldsymbol{D} \cdot \boldsymbol{\delta} \tag{3.28}$$

where $\boldsymbol{D}$ is specified as:

$$D_{ij} = \begin{cases} K_p \bar{\delta}_{ij}, & \alpha \leq \delta_m^0 \\ \bar{\delta}_{ij} \left[(1-d) K_p + d K_p \dfrac{\langle -\delta_3 \rangle}{-\delta_3} \bar{\delta}_{i3} \right], & \delta_m^0 < \alpha < \delta_m^f \\ \bar{\delta}_{i3} \bar{\delta}_{3j} \dfrac{\langle -\delta_3 \rangle}{-\delta_3} K_p, & \alpha \geq \delta_m^f \end{cases} \tag{3.29}$$

To apply the equations to user subroutine for material, one need to set the values for strengths σ_0 and τ_0, penalty stiffness K_p, and B-K parameter η. The critical energy release rate G_{Ic} and G_{IIc} can be obtained from experiments.

In the linear softening model described before, the layer strengths are at first user-specified values as well as the penalty stiffness. Based on the fracture energy release rate from experiment results, the displacement of total decohesion, δ_f, is calculated every time with respect to the mode-mixing ratio. Another possible thought is to fix the displacement/separation, δ_f, such that the cohesive elements will be opened when their separation between the upper and lower faces reach a certain user-assumed value. So a modified linear softening cohesive law is proposed here.

By introducing a decohesion to propagation separation ratio r (or just separation ratio), together with the decohesion displacement δ_f, the displacement for propagation onset is now:

$$\delta_0 = \frac{\delta_f}{r} \tag{3.30}$$

The strength of the cohesive layer was set to a fixed value before. Now it will be expressed as:

$$\sigma_0 = \frac{2G_{\mathrm{Ic}}}{\delta_f} \tag{3.31}$$

Thus, the penalty stiffness will change from 10^6 N/mm^3 which is from the linear cohesive model before to:

$$K_p = \frac{2G_{\mathrm{c}}r}{(\delta_f)^2} \tag{3.32}$$

The main difference of this model compared to the original linear softening model is that the normal stress in the cohesive element does not determine the failure of the layer. The propagation is defined solely geometrically (separation). These are the changes that are made to the pure mode formulation.

The changes in pure mode formulation will also affect the mixed-mode formulation. This change is illustrated in Figure 3.4.

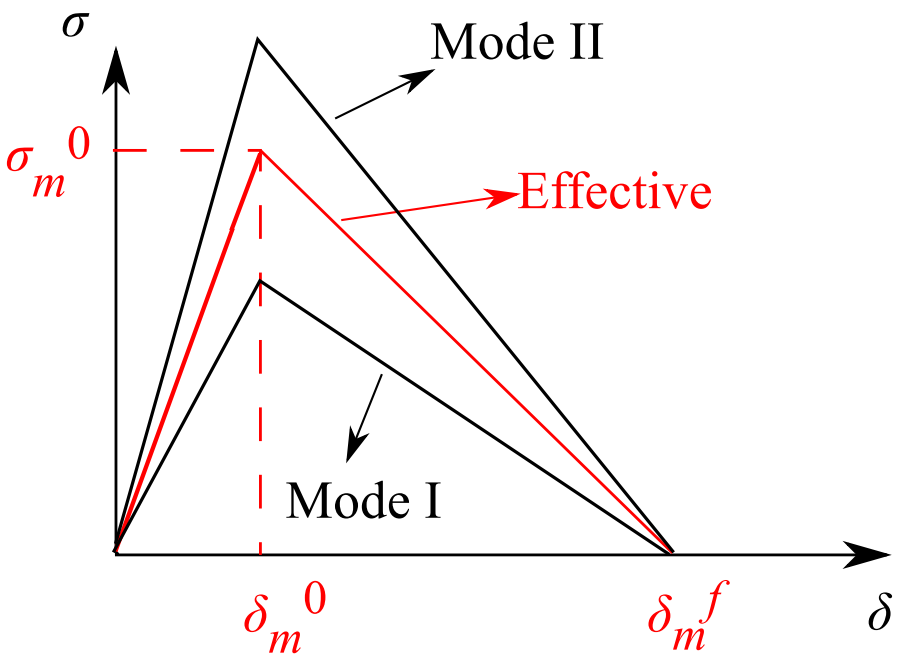

Figure 3.4. Modified linear cohesive law mixed-mode behavior

With the definition of effective displacement in Eq. (3.17), it can be seen that:

$$\delta_m^0 = \delta_0 \tag{3.33}$$

which is true for all mode mixities. Substitute Eq. (3.33) to the left side of Eq. (3.19), the equation still holds, which means this model still satisfy the quadratic failure criterion. Define an effective penalty stiffness K_{pm}, which is:

$$K_{pm} = \frac{\sigma_m^0}{\delta_m^0} = \frac{\sigma_m^0}{\delta_0} \tag{3.34}$$

where σ_m^0 is defined similar to Eq. (3.31):

$$\sigma_m^0 = \frac{2G_c}{\delta_m^f} = \frac{2G_c}{\delta_f} \tag{3.35}$$

Here, the energy release rate terms in normal and shear mode will be defined directly using δ_f:

$$\begin{aligned} G_{\mathrm{I}} &= \frac{\sigma_0 \delta_f}{2} \\ G_{\mathrm{shear}} &= \frac{\tau_0 \delta_f}{2} \end{aligned} \tag{3.36}$$

Substitute Eq. (3.35) and Eq. (3.36) back to the B-K criterion Eq. (3.22), solve for σ_m^0:

$$\sigma_m^0 = \begin{cases} \dfrac{2}{\delta_f}\left[G_{\mathrm{Ic}} + (G_{\mathrm{IIc}} - G_{\mathrm{Ic}})\left(\dfrac{\beta^2}{1+\beta^2}\right)^{\eta}\right], & \delta_3 > 0 \\ \dfrac{2G_{\mathrm{IIc}}}{\delta_f}, & \delta_3 \leq 0 \end{cases} \tag{3.37}$$

Thus, from Eq. (3.34), one can get the value for the effective stiffness for any mode-mixity in terms of σ_m^0. Then the constitutive relation will be the same as Eq. (3.16) except that K_p is replaced with K_{pm}. This completes the formulation for modified linear softening law, which still follows the quadratic failure criterion and B-K mixed-mode criterion. This modified linear cohesive law is used to demonstrate that a modification can be made to cohesive laws that already exist and tested in simulation. At the same time, it will help with the derivation of power-law cohesive law in the next section.

3.2.3 Power-Law Cohesive Law

In the linear cohesive model, the description for damage parameter d is Eq. (3.27). This parameter is related with the displacement of damage propagation and total decohesion. It can be changed into a power law description which is [20]:

$$d = 1 - \left(\frac{\delta_m^f - \alpha}{\delta_m^f - \delta_m^0}\right)^n \tag{3.38}$$

This is called power-law softening which has a smoother traction-separation relation since the formulation has higher orders. Typically, the area below the curve is still the critical fracture energy release rate, which makes the formulation of such models harder and the previous procedures can not be completely reproduced in this case. A

power-law cohesive model based on the modified linear cohesive law is then proposed here.

The idea here is to use the modified linear cohesive law as an estimation of the shape of power-law softening traction-separation relation. By making a higher order curve, it is expected that the load-displacement curve quality can be improved. Since the modified linear cohesive law shows some insufficiency in dissipated energy, the power law model can be modified such that the cohesive layer strength is the same as the modified linear softening model above. This power law's pure mode behavior is shown in Figure 3.5.

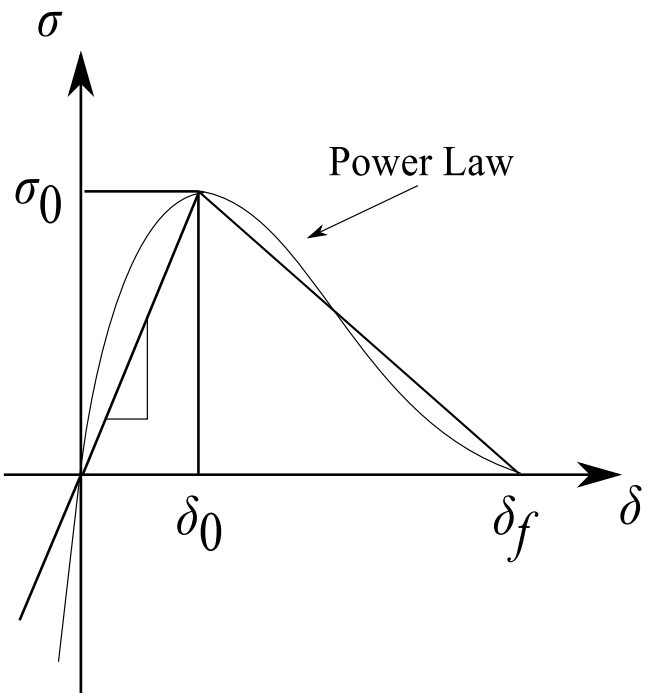

Figure 3.5. Power-law and linear cohesive laws

Here $n = 2$ is chosen so that the formulation will be easy to derive. In the elastic region, the traction-separation relation has a second order polynomial, which is:

$$\sigma = a_2\delta^2 + a_1\delta, \qquad \delta \leq \delta_0 \tag{3.39}$$

In the softening part, the relation is governed by the definition of d in Eq. (3.38). Thus:

$$\sigma = (1-d)\sigma_0 = (1-d)\frac{2G_c}{\delta_f} \tag{3.40}$$

All other formulas of pure mode are the same as derived in the last section.

Since the modified linear cohesive law already incorporates mixed-mode behavior on the basis of quadratic failure criterion and B-K mixed-mode criterion, the formulation in terms of power law will basically focus on the formulation of constitutive relations.

The traction-separation relation consists of two second-order polynomial parts. This model will still use the same definitions for effective displacement and mode-mixing ratio that are formulated in Eq. (3.17) and Eq. (3.18). Since δ_0 is now user-defined parameter, now setting (just as the linear softening model sets σ_0 and τ_0 to certain values):

$$\delta_3^0 = \delta_2^0 = \delta_1^0 = \frac{\delta_f}{r} \tag{3.41}$$

For the part before damage onset, the power law softening exhibits a non-linear elastic property, such that the curve fits better with the softening part and do not create a sharp change at the top. In this case, it is more likely to fit with real situations. Since now n is set to 2, there is still a discontinuity in the slope of the traction-separation relation. Continuity of the slope in the traction-separation relation can improve the quality of load-displacement curve around the critical traction, though there is no need to enforce it since the values of critical traction and separation will not be affected by it. Here, the relation between traction and separation in the damage initiation part is:

$$\sigma = a_2\delta_3^2 + a_1\delta_3, \qquad \delta_3 \leq \delta_3^0 \tag{3.42}$$

where a_1 and a_2 are unknown variables. By solving the equations:

$$\begin{cases} 2a_2\delta_3^0 + a_1 = 0 \\ a_2(\delta_3^0)^2 + a_1\delta_3^0 = \sigma_0 = \dfrac{2G_{\mathrm{Ic}}}{\delta_f} \end{cases} \tag{3.43}$$

expressions for traction can be get:

$$\sigma = -\frac{2G_{\mathrm{Ic}}r^2}{(\delta_f)^3}(\delta_3)^2 + \frac{4G_{\mathrm{Ic}}r}{(\delta_f)^2}\delta_3, \qquad \delta_3 \leq \delta_3^0 \tag{3.44}$$

Likewise, for the shear directions:

$$\tau_i = -\frac{2G_{\mathrm{IIc}}r^2}{(\delta_f)^3}(\delta_i)^2 + \frac{4G_{\mathrm{IIc}}r}{(\delta_f)^2}\delta_i, \qquad i = 1, 2, \qquad 0 \leq \tau_i \leq \delta_3^0 \tag{3.45}$$

But for $\tau_i < 0$, the curve should be the same as in the positive region, so:

$$\tau_i = \frac{2G_{\mathrm{IIc}}r^2}{(\delta_f)^3}(\delta_i)^2 + \frac{4G_{\mathrm{IIc}}r}{(\delta_f)^2}\delta_i, \qquad i = 1, 2, \qquad -\delta_3^0 \leq \tau_i < 0 \tag{3.46}$$

For the shear strength of cohesive layer, there is:

$$\tau_0 = \frac{2G_{\mathrm{IIc}}}{\delta_f} \tag{3.47}$$

Since the cohesive law in terms of separation for the three modes are considered the same, there is:

$$\begin{aligned} \delta_m^0 &= \delta_3^0 \\ \delta_m^f &= \delta_3^f \end{aligned} \tag{3.48}$$

The parameters for damage onset and total decohesion are based on the modified linear softening law, this makes the formulation in terms of mode-mixing behavior a lot simpler. Because there is no need to make the area below the curve exactly the same as critical fracture energy release rate, but rather an approximation of the energy release rate value. For the constitutive relation in the softening part:

$$\begin{aligned} \sigma &= (1-d)\sigma_0 = (1-d)\frac{2G_{\mathrm{Ic}}}{\delta_f} \\ \tau &= (1-d)\tau_0 = (1-d)\frac{2G_{\mathrm{IIc}}}{\delta^f} \end{aligned} \tag{3.49}$$

where d is defined in Eq. (3.38). Also, the loading and unloading conditions are important information in this case. Because when unloading, the stress-strain relation will follow a linear elastic path from current stress-strain condition to the origin. The loading function F is defined as:

$$F = \frac{\langle \delta_m - \alpha \rangle}{\delta_m - \alpha} \tag{3.50}$$

where $F = 1$ denotes loading, and $F = 0$ means unloading/reloading/no loading. Same as the mixed mode formulation, the power law is governed only by one state variable α, which is the maximum separation reached in loading history and is used to track the damage evolution state.

Now the power-law based on the modified linear cohesive law is derived, the illustration of mixed-mode behavior is in Figure 3.6, where all the different mode-mixity states between Mode I and Mode II will share the same effective displacement for propagation onset and total decohesion.

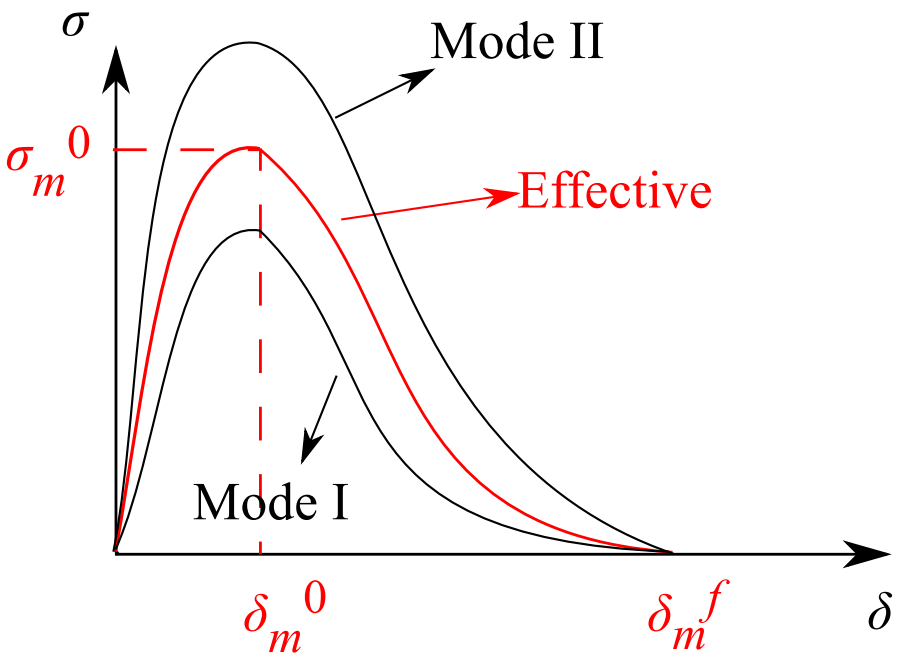

Figure 3.6. Power law mixed-mode behavior

3.3 VUMAT

A user subroutine, VUMAT, provides the possibility for users to give selected sections of the finite element model modified constitutive relations. It defines material properties for ABAQUS/Explicit. In this work, the cohesive layer needs VUMAT program to have user-defined property. The most important formulation derived in this chapter is the equations for constitutive relations which are Eq. (3.16) and Eq. (3.44) to Eq. (3.46).

At the start of each increment for each integration point of every element, VUMAT will be called to provide these constitutive relations. Based on the current displacement and traction state of the element, the program can determine the stress state at the end of this increment by adding a displacement increment to current displacement, and determining new stresses using the formulation derived in this chapter. At the end of the program, the stresses are passed back to ABAQUS/Explicit. For a certain integration point that has reached displacement for total decohesion, the stresses will be set to 0 at every following increment such that this point act as if it's completely damaged. There is no need to actually delete the element that has all integration points damaged. At the same time, the state variable α which records the maximum effective displacement reached in history will be updated for each integration point every time VUMAT is called. This helps decide the extent of damage for that integration point.

All three VUMAT programs for the three cohesive laws discussed in this chapter are provided in Appendix A. The tests carried out in the following chapter all have these programs along with ABAQUS.

4. Results and Discussion

In this chapter, finite element model parameters will be properly selected using the present approach. The convergence and accuracy of the finite element model, together with the predetermined model parameters, will be validated. The illustrative linear and power-law cohesive laws will be programmed with VUMAT and adapted to the finite element model. The finite element model capability of handling various interfacial models will be demonstrated.

4.1 Experimental Parameters

The test specimen is a 2-ply plate-shaped UD AS4/PEEK carbon-fiber reinforced composite laminate with the material parameters listed in Table 4.1, where σ_0 and τ_0 denote the layer strength in normal and shear directions, respectively. The laminate is considered uni-directional in terms of fiber orientation. Recall the specimen dimensions in Table 4.2.

Table 4.1. Material properties

E_{11} (GPa)	$E_{22} = E_{33}$ (GPa)	$G_{12} = G_{13}$ (GPa)	G_{23} (GPa)	$\nu_{12} = \nu_{13}$
122.7	10.1	5.5	3.7	0.25
ν_{23}	G_{Ic} (KJ/m^2)	G_{IIc} (KJ/m^2)	σ_0 (MPa)	τ_0 (MPa)
0.45	0.969	1.719	80	100

In the present approach, the specimen is meshed using 8-node brick elements, with 408 elements in the longitudinal direction, 8 elements in the thickness direction, and

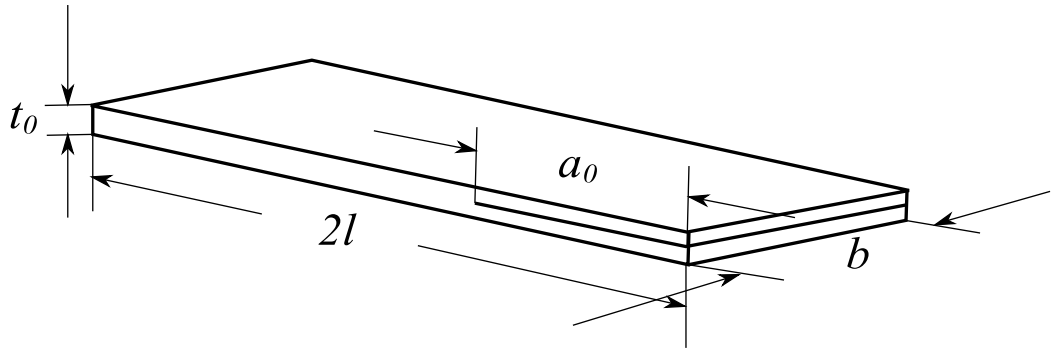

Figure 4.1. Specimen dimensions

Table 4.2. Specimen geometry (a duplication of Table 2.2)

	length ($2l$)	width (b)	thickness (t_0)	crack length (a_0)
dimension (mm)	102	25.4	3.12	33

5 elements in the width direction. The cohesive layer is meshed with zero-thickness 8-node cohesive elements which are embedded with cohesive laws defined in VUMAT.

4.2 Model Parameters

The default value of constitutive thickness is 1 in ABAQUS finite element analysis such that the nominal strains of the cohesive layer will be equal to the relative separation displacements. By using different thickness values, the DCB test is simulated, and the force-displacement results are shown in Figure 4.2.

The curve quality is not affected much by different values of constitutive thickness, though the computation cost may vary. The corresponding numbers of increments for each thickness value are listed in Table 4.3. Thus, a larger constitutive thickness can reduce, though not significantly, the computation cost of the simulation of delamination. This aspect can be considered when modeling the delamination of such

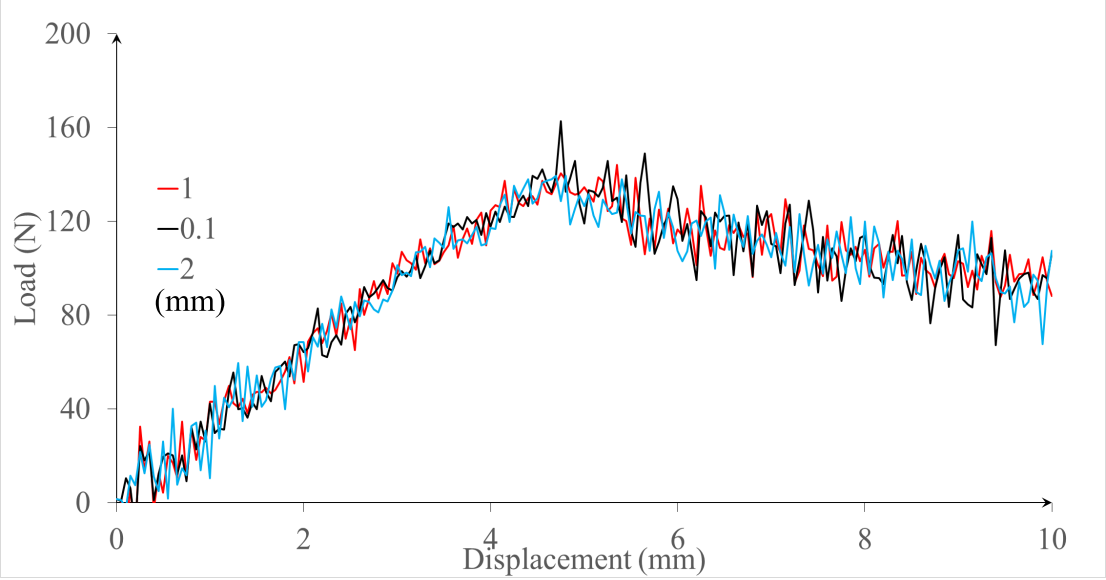

Figure 4.2. Load-displacement curves for different values of constitutive thickness (DCB test)

specimen with cohesive layer. From the data in Table 4.3, setting the thickness to 1 mm is good enough in terms of reducing the computation cost.

Table 4.3. Computational cost for different values of constitutive thickness

thickness (mm)	0.1	0.5	1	2
increments ($\times 10^5$)	18.6	7.6	6.6	6.5

The default viscosity parameter in ABAQUS is set to 0.06. By adjusting this number, the simulations give different results in Figure 4.3, and the computation cost shown in Table 4.4. While Figure 4.3 shows that different viscosity terms may not cause very different curves in the linear part of load-displacement relation. In the propagating part, higher viscosity value does help with better results, and with higher computation cost at the same time. The value 0.3 here is sufficient enough for this problem and will be adopted.

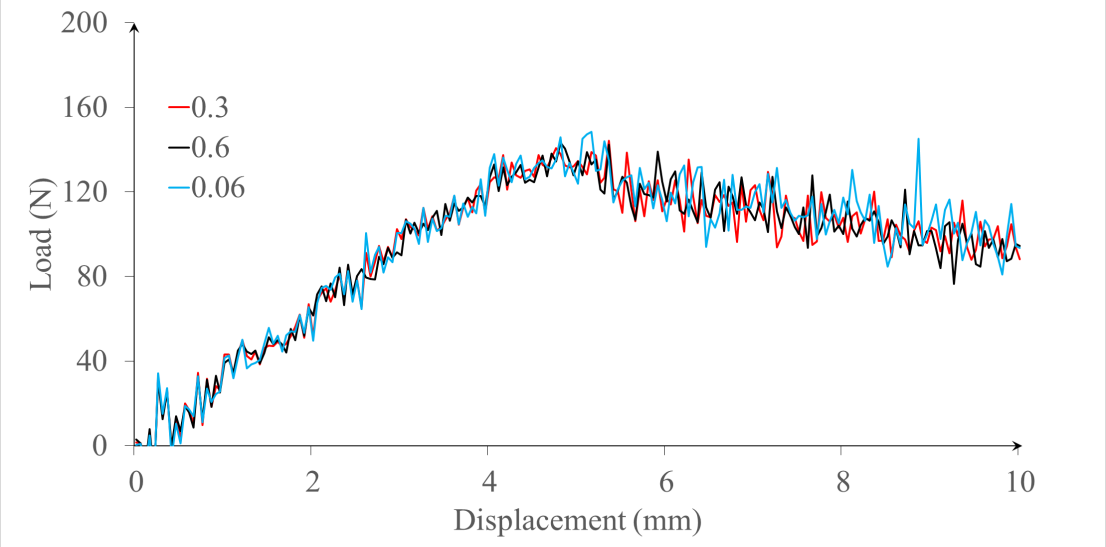

Figure 4.3. Load-displacement curves for different values of viscosity (DCB test)

Table 4.4. Computational cost for different values of viscosity

viscosity	0.06	0.3	0.6
increments ($\times 10^5$)	5.2	6.6	8.7

Simulations using the same model set-up except for penalty stiffness are carried out, the load-displacement curves are shown in Figure 4.4.

Table 4.5. Kinetic energy ratio for different values of penalty stiffness

K_p(N/mm^3)	10^4	10^5	10^6
kinetic ratio (%)	429	183	0.46

The load-displacement curve seems not affected by the change in density stiffness, given that the critical fracture energy release rate of material remains the same. Though further increase the stiffness to 10^7 N/mm^3 does cause very unrealistic solutions. Although lower penalty stiffness does not show major difference in this test, it

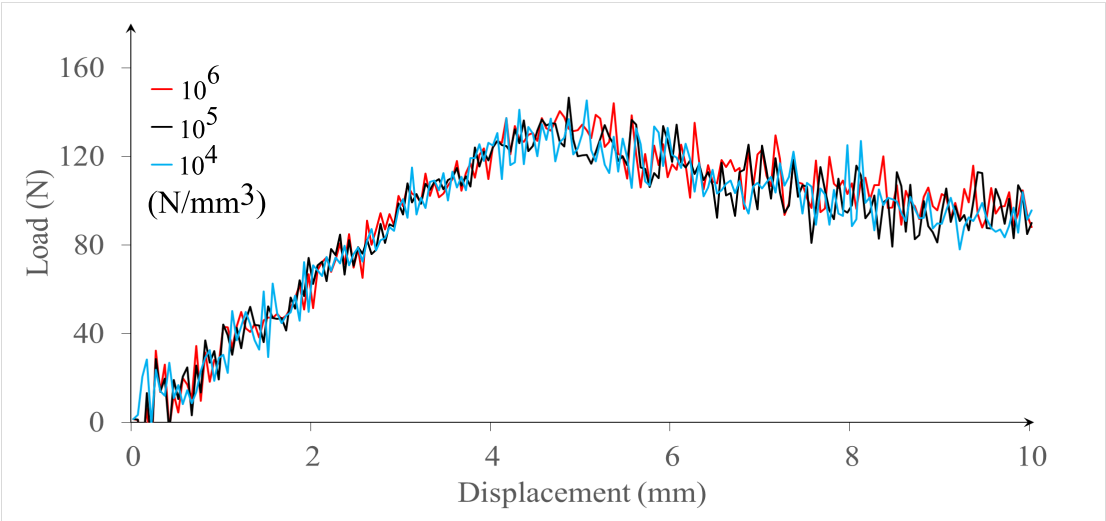

Figure 4.4. Load-displacement curves for different values of penalty stiffness (DCB test)

is still worth noticing that this is a dynamic analysis which does not have as many convergence problems as simulations carried out using static analysis. The kinetic energy to total strain energy ratio of different penalty stiffnesses is show in Table 4.5. It is obvious that when penalty stiffness is small, the process tends to get very dynamic, from which we can infer that there will be convergence problems during static analysis. A stiffness set to the same order magnitude as 10^6 N/mm^3 would be best for analysis.

Simulations using the default set-up except for the density of both the bulk material and the cohesive layer are carried out, and the results shown in Figure 4.5. Since the kinetic energy to total strain energy ratio is important here, they are also compared in Table 4.6.

It can be seen from Figure 4.5 that, by decreasing the value of density, load-displacement curve becomes smoother, which shows the process becomes less and less dynamic with the decrease in density. But at the same time, from Table 4.6, a higher computation cost (increment number) is observed. A choice with density

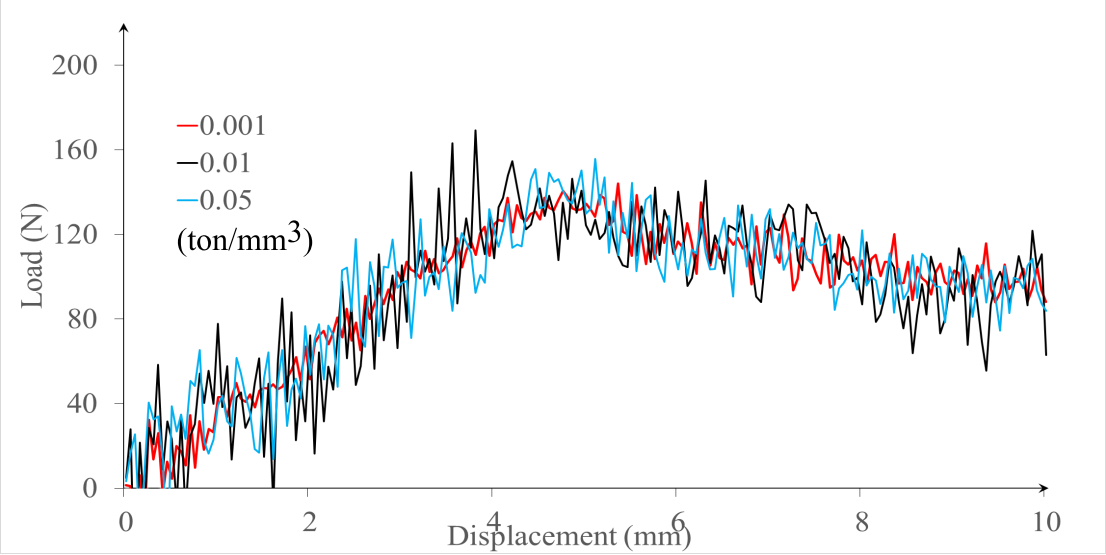

Figure 4.5. Load-displacement curves for different values of density (DCB test)

Table 4.6. Computational cost and kinetic ratio for different values of density

density (ton/mm^3)	0.001	0.005	0.01	0.5
increments ($\times 10^5$)	6.6	2.9	2.1	0.3
kinetic ratio (%)	0.46	3.40	4.96	2800

value 0.001 ton/mm^3 is made after this analysis, ensuring reasonable computation cost, and guaranteeing accuracy of solution at the same time.

A typical quasi-static problem is suggested to have its kinetic energy to total strain energy ratio smaller than 5%. This is used as a criterion in this work to determine the validity of simulation set-up. Decreasing the load rate can yield smoother traction vs separation curve. For the DCB test, setting the maximum displacement to 10 mm, different load rates can be achieved by using different step time. The simulations are carried out based on the default specimen set-up, results of which is shown in Figure 4.6.

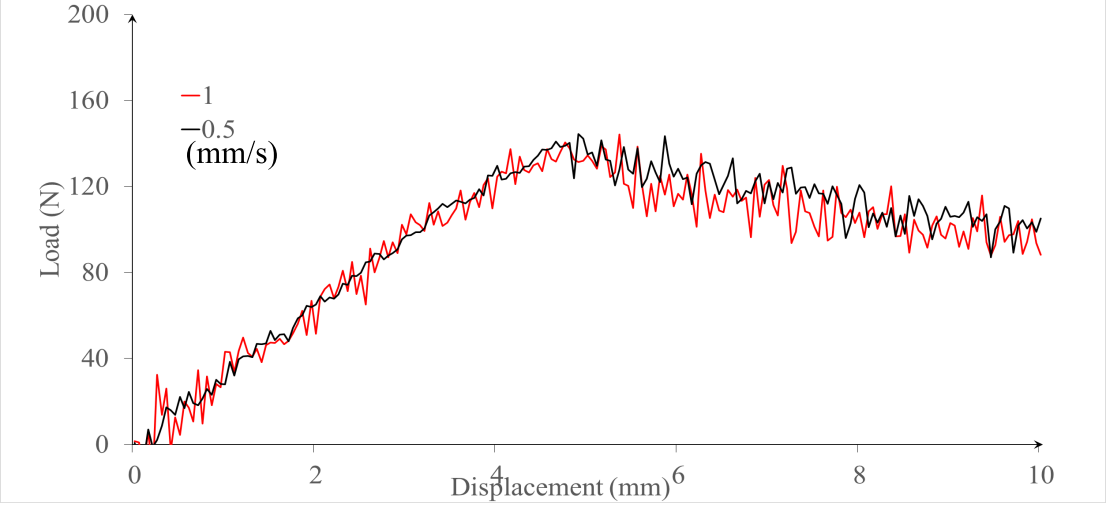

Figure 4.6. Load-displacement curves for different values of load rate (DCB test)

Though using different load rates does not show obvious difference between results, with lower load rate, the energy dissipated seems to be more than with higher load rate. Of course, lower load rate can generate better load-displacement curves. For the following case, a load rate of 0.5 mm/s is used for the sake of accuracy of the simulations.

4.3 Validation and Demonstrations

4.3.1 Validation of Model Parameters

After all the results from different model parameters worked out, a experimental setup for the finite element framework is established. Some of the parameters discussed above are then determined. Table 4.7 shows these parameters.

Based on the linear cohesive law and all the parameters above, the DCB test is simulated and compared with experiment results as well as numerical results from [8], where the simulation model features shell elements instead of block elements in this work (Figure 4.7).

Table 4.7. Finite element model parameters

density (ton/mm^3)	viscosity	penalty stiffness (K_p) (N/mm^3)
0.001	0.03	10^6
load rate (mm/s)	constitutive thickness (mm)	
0.5	1	

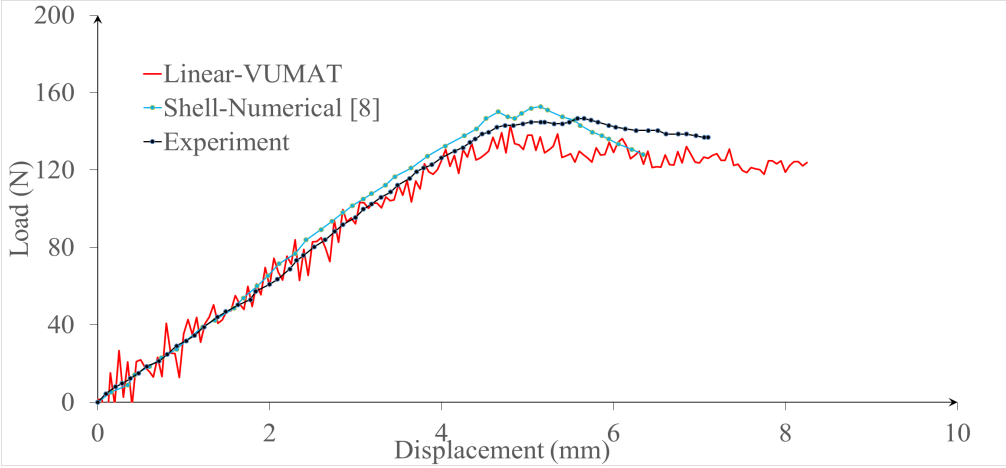

Figure 4.7. Load-displacement curves for linear cohesive law (DCB test)

The curve for linear cohesive law in Figure 4.7 is not smooth enough to show details because of the relatively coarse mesh. It can still be seen that a good agreement is reached between the block-element numerical result and the experiment result, since the numerical result using shell elements shows an obvious load drop after the start of propagation. Though the load-displacement curve obtained here is still below the experiment curve, which may be caused by over-estimation of layer strength and insufficient load rate. A comparison between original linear cohesive law and the modified linear cohesive law is shown in Figure 4.8 also with DCB test.

From the figure we can see that the modified law shows even lower energy dissipation in the propagation area than the linear softening model, but also yield better

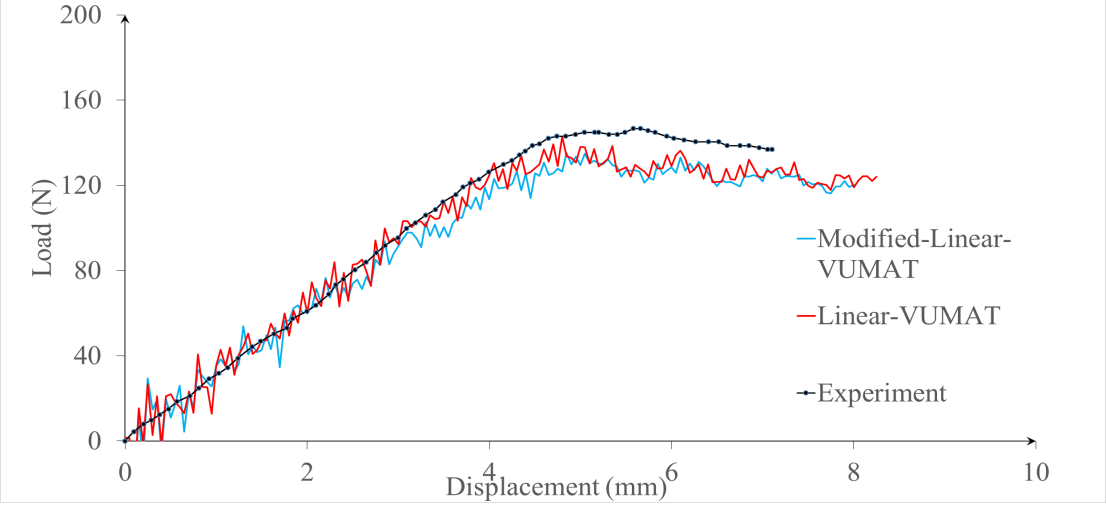

Figure 4.8. Comparison of load-displacement curves between linear cohesive law and modified linear cohesive law (DCB test)

curve quality with the same specimen used in the original test. With the parameters in the model further discussed, we can still make the model fit with experiment results more.

MMB tests are carried out based on modified linear cohesive law and power law described in the previous chapter and by applying two forces at the end and the middle of the specimen, respectively, and with mode-mixities 20%, 50% and 80%. The method to compute the ratios between the forces based on different mode-mixities can be found in Ref. [8]. To compute the load-point displacement from the displacements at the end and the middle of the specimen [21]:

$$w = \frac{c+l}{l} w_m - \frac{c}{l} w_e \tag{4.1}$$

where w, w_m, and w_e are displacements at the load point, the middle, and the end of the specimen, respectively. So the displacement results from ABAQUS can be computed into the displacement at the load point with Eq. (4.1). Using VUMAT embedded with linear cohesive law, the MMB tests are simulated. Results are in Figures 4.9, 4.10, and 4.11. The curves in these three figures shows a good agreement

among results from linear VUMAT program, experiment results, and results from models using shell elements, though both numerical methods seem to underestimate the critical load. The linear cohesive law causes unstable curves after reaching the critical load, especially in the 20% test. This is because of the change in slope of the traction-separation law at the critical load. The present finite element framework and VUMAT for linear cohesive law can be validated.

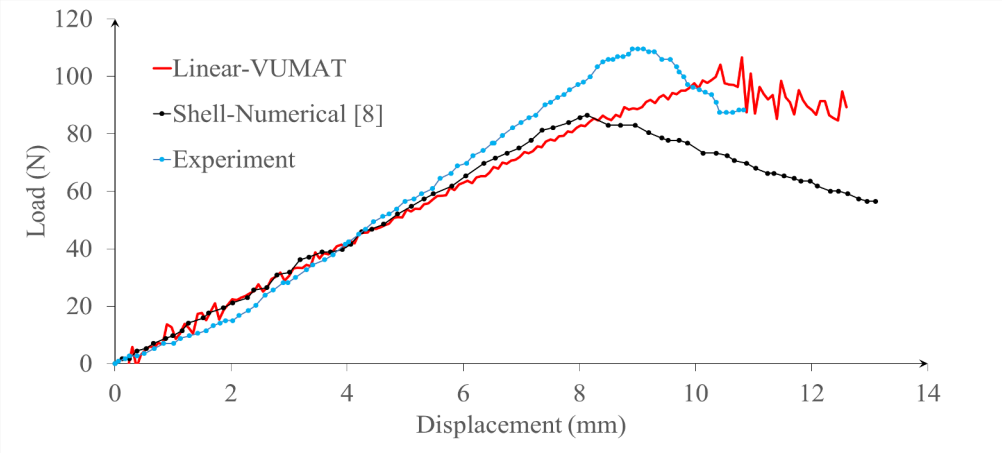

Figure 4.9. Load-displacement curves for linear cohesive law with a mode-mixing ratio of 20% (MMB test)

4.3.2 Demonstrations

Other than linear cohesive law, power law can also be used to simulate the MMB test with the present finite element framework. Using VUMAT embedded with power law, the MMB tests are simulated. Results are in figures 4.12, 4.13, and 4.14.

Generally all the curves of different mode-mixing ratios agree well with the experiment results. Though the curve for mode-mixing ratio 80% seems to underestimate the critical load. There may be some causes for this problem such as the selection of material property within the cohesive law, which can be adjusted with further parametric study. The curves of power-law cohesive law are generally postponed in terms of damage softening. Because the traction-separation law in power law has

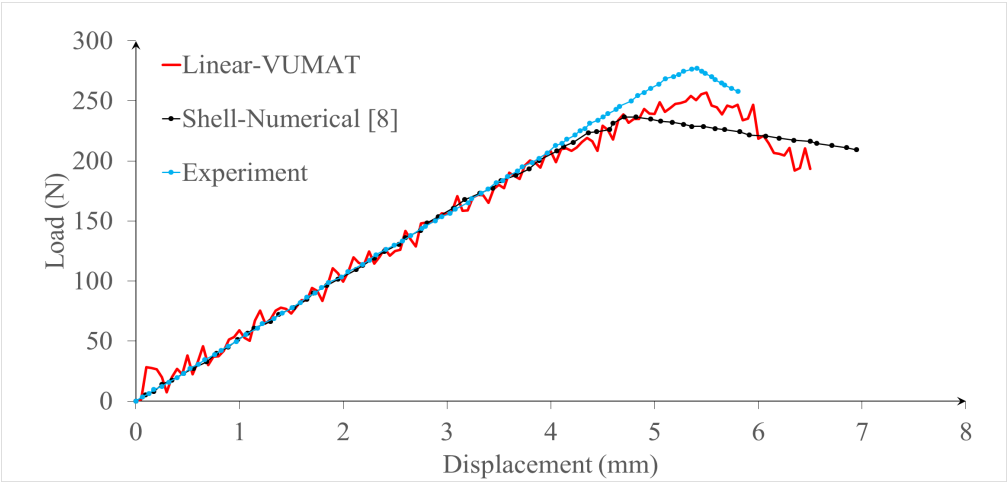

Figure 4.10. Load-displacement curves for linear cohesive law with a mode-mixing ratio of 50% (MMB test)

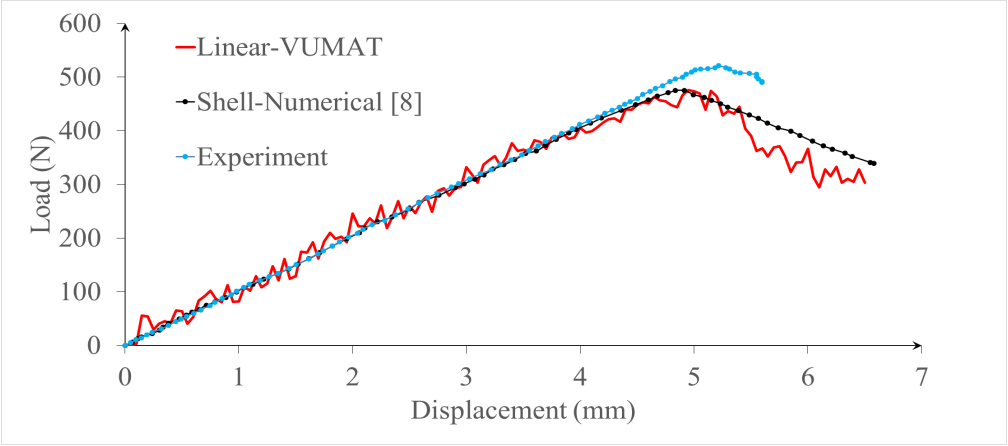

Figure 4.11. Load-displacement curves for linear cohesive law with a mode-mixing ratio of 80% (MMB test)

smoother curve, which makes dissipated energy less than linear cohesive law. This will potentially postpone the softening.

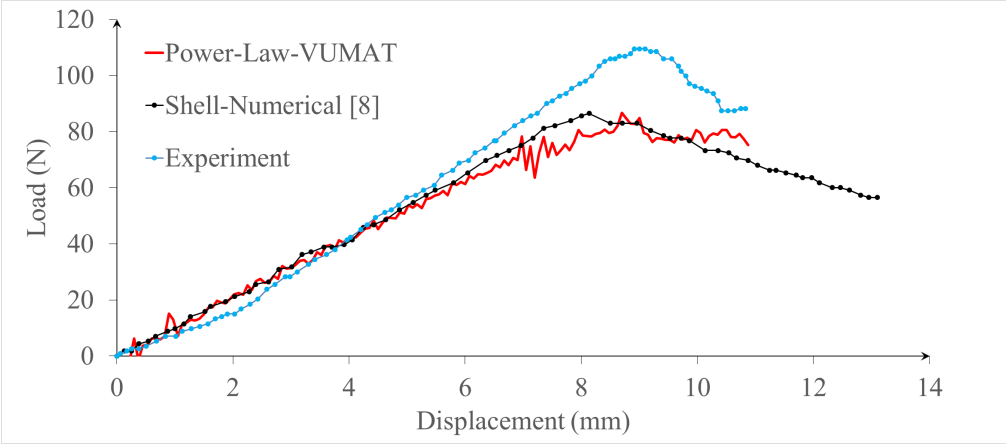

Figure 4.12. Load-displacement curves for power law with a mode-mixing ratio of 20% (MMB test)

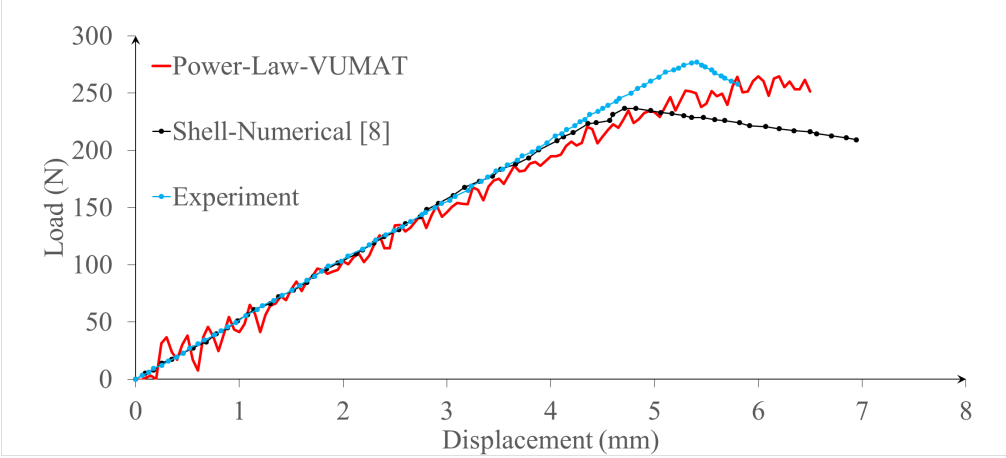

Figure 4.13. Load-displacement curves for power law with a mode-mixing ratio of 50% (MMB test)

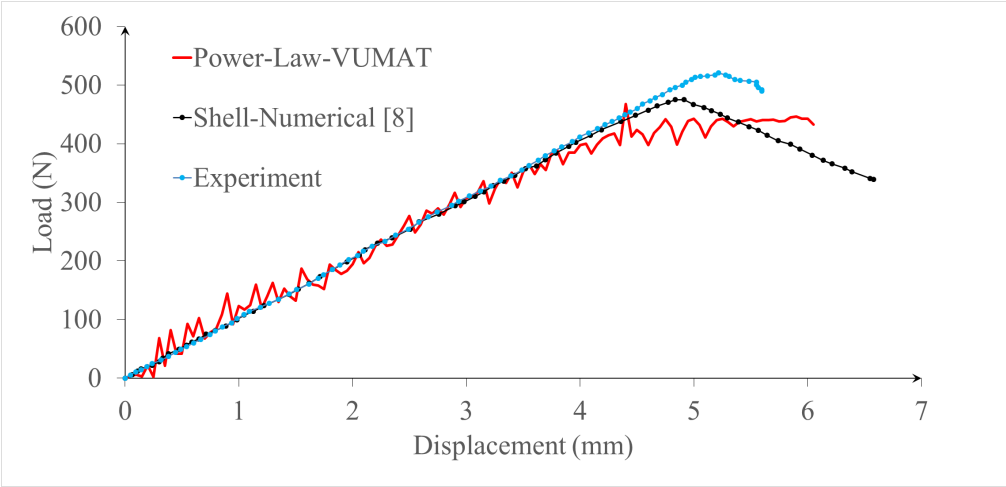

Figure 4.14. Load-displacement curves for power law with a mode-mixing ratio of 80% (MMB test)

5. Conclusions

A finite element framework, which consists of properly selecting time integration scheme (explicit dynamic), viscosity, load rate and mass scaling, is developed to yield converged and accurate results. Two illustrative cohesive laws (linear and power law) are programmed with a user-defined material subroutine VUMAT for ABAQUS/Explicit, and implemented into the finite element framework.

The following findings can be obtained from the results:

1. finite element model parameters can be properly determined using the present approach;

2. the finite element model, together with the predetermined model parameters, is found to be capable of producing converged and accurate results;

3. the illustrative linear and power-law cohesive laws can be conveniently implemented in the finite element model with VUMAT;

4. the finite element model, embedded with the illustrative cohesive laws, is found to be capable of handling various interfacial models.

The following conclusions can be drawn from the above findings:

1. more sophisticated laminate properties and cohesive laws can be implemented into the present approach;

2. the present approach can be further extended to handle the interfacial debonding in many other heterogeneous materials (e.g., fiber and particle reinforced composites);

3. although the present approach is developed with ABAQUS, it is also amendable to many other finite element codes which has cohesive element and with the capability of incorporating user defined material models.

REFERENCES

REFERENCES

[1] J. R. Reeder and Jr. J. H. Crews. Mixed-mode bending method for delamination testing. *AIAA Journal*, 28(7):1270–1276, 1990.

[2] E. F. Rybicki and M. F. Kanninen. A finite element calculation of stress intensity factors by a modified crack closure integral. *Engineering Fracture Mechanics*, 9:931–938, 1977.

[3] K. N. Shivakumar, P. W. Tan, and Jr. J. C. Newman. A virtual crack-closure technique for calculating stress intensity factors for cracked three dimensional bodies. *International Journal of Fracture*, 36:R43–R50, 1988.

[4] K. F. Nilsson. On growth of crack fronts in the dcb test. *Composite Engineering*, 3(6).

[5] W. Cui and M. R. Wisnom. A combined stress-based and fracture-mechanics-based model for predicting delamination in composites. *Composites*, 24(6):467–474, 1993.

[6] E. B. Glennie. A strain-rate dependent crack model. *Journal of the Mechanics and Physics of Solids*, 19:255–272, 1971.

[7] P. P. Camanho, C. G. Dávila, and S. T. Pinho. Fracture analysis of composite co-cured structural joints using decohesion elements. *Journal of Fatigue and Fracture of Engineering Materials and Structures*, 27:745–457, 2004.

[8] P. P. Camanho, C. G. Dávila, and M. F. de Moura. Numerical simulation of mixed-mde progressive delamination in composite materials. *Journal of Composite Materials*, 37(16), 2003.

[9] A. Turon, P. P. Camanho, J. Costa, and J. Renart. Accurate simulation of delamination growth under mixed-mode loading using cohesive elements: Definition of interlaminar strengths and elastic stiffness. *Journal of Composite Structures*, 92:1857–1864, 2010.

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

[11] Abaqus analysis user's manual. 2011.

[12] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15:527–551, 1979.

[13] I. Lapczyk and J. A. Hurtado. Progressive damage modeling in fiber-reinforced materials. *Composites*, 38:2333–2341, 2007.

[14] Z.Hashin. Failure criteria for unidirectional fiber composites. *Journal of Applied Mechanics*, 47:329–334, 1980.

[15] Y. Mi, M. A. Crisfield, and G. A. O. Davies. Progressive delamination using interface elements. *Journal of Composite Materials*, 32(14).

[16] L. Daudeville, O. Allix, and P. Ladevèze. Delamination analysis by damage mechanics: Some applications. *Composite Engineering*, 5(1):17–24, 1995.

[17] C. G. Dávila, P. P. Camanho, and M. F. de Moura. Mixed-mode decohesion elements for analysis of progressive delamination. *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (Seatle, Washington, U.S.A.)*, 2001.

[18] C. G. Dávila and P. P. Camanho. Decohesion elements using two and three-parameter mixed-mode criteria. *American Helicopter Society International Structures Specialists' Meeting (Williamsburg, Virginia, U.S.A.)*, 2001.

[19] M. L. Benzeggagh and M. Kenane. Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus. *Journal of Composites Science and Technology*, 56:439–449, 2003.

[20] J. Mosler and I. Scheider. A thermodynamically and variationally consistent class of damage-type cohesive models. *Journal of the Mechanics and Physics of Solids*, 59:1647–1668, 2011.

[21] C. Balzani and W. Wagner. An interface element for the simulation of delamination in unidirectional fiber-reinforced composite laminates. *Journal of Engineering Fracture Mechanics*, 75:2597–2615, 2008.

APPENDIX

A. User Subroutine Programs

In this appendix, the FORTRAN codes for ABAQUS VUMAT are presented.

A.1 Linear Softening Model

```
      subroutine linearsoftening(
c       Read Only Variables
     1  nblock,ndir,nshr,nstatev,nfieldv,nprops,lanneal,
     2  stepTime,totalTime,dt,cmname,coordMp,charLength,
     3  props,density,strainInc,relSpinInc,
     4  tempOld,stretchOld,defgradOld,fieldOld,
     5  stressOld,stateOld,enerInternOld,enerInelasOld,
     6  tempNew,stretchNew,defgradNew,fieldNew,
c       Write Only Variables
     7  stressNew,stateNew,enerInternNew,enerInelasNew)
c
      include 'vaba_param.inc'
c
      dimension props(nprops),density(nblock),coordMp(nblock),
     1  charLength(nblock),strainInc(nblock,ndir+nshr),
     2  relSpinInc(nblock,nshr),tempOld(nblock),
     3  stretchOld(nblock,ndir+nshr),defgradOld(nblock,ndir+nshr+nshr),
     4  fieldOld(nblock,nfieldv),stressOld(nblock,ndir+nshr),
     5  stateOld(nblock,nstatev),enerInternOld(nblock),
     6  enerInelasOld(nblock),tempNew(nblock),
     7  stretchNew(nblock,ndir+nshr),defgradNew(nblock,ndir+nshr+nshr),
```

```
     8  fieldNew(nblock,nfieldv),stressNew(nblock,ndir+nshr),
     9  stateNew(nblock,nstatev),enerInternNew(nblock),
     1  enerInelasNew(nblock)
c
      character*80 cmname
c
      real nstressa,sstressa,KK,Gic,Giic,nstraina,sstraina
      real alpha,beta,nstrainf,sstrainf,strainEfff,eta,d
      integer i,j,k,l
      real stressAbs,strainAbs
      real strainNew(ndir+nshr),strainEff,strainEffa
      real rshear
      real DD(3,3),Ic(3,3),II(3,3)
c     Layer strengths, penalty stiffness, fracture energy,
c         B-K parameter are defined here
      parameter (nstressa=80,sstressa=100,KK=100000,
     1  Gic=0.969,Giic=1.719,eta=2.284)
c     Calculate displacement for propagation onset
      nstraina=nstressa/KK
      sstraina=sstressa/KK
      do i=1,nblock
c     Calculate <normal strength>
        if (stressOld(i,1).gt.0) then
          stressAbs=stressOld(i,1)
        else
          stressAbs=0
        end if
c     Calculate new displacement
        do j=1,ndir+nshr
```

```
          strainNew(j)=stateOld(i,j+1)+strainInc(i,j)
          stateNew(i,j+1)=strainNew(j)
        end do
c     Calculate <normal displacement>
        if (strainNew(1).gt.0) then
          strainAbs=strainNew(1)
        else
          strainAbs=0
        end if
c     Calculate effective displacement and shear displacement
        strainEff=strainAbs**2
        rshear=0
        do j=1,nshr
          rshear=rshear+4*strainNew(ndir+j)**2
        end do
        strainEff=strainEff+rshear
        strainEff=sqrt(strainEff)
        rshear=sqrt(rshear)
c     Determine maximum displacement reached in history
        alpha=stateOld(i,1)
        if (strainEff.gt.alpha) then
          alpha=strainEff
        end if
        stateNew(i,1)=alpha
c     Calculate displacement mode mixity
        beta=rshear/strainNew(1)
c     Calculate effective propagation onset displacement
        if (strainNew(1).gt.0) then
          strainEffa=nstraina*sstraina
```

```
     1      *sqrt((1+beta**2)/(sstraina**2+(beta*nstraina)**2))
            else
              strainEffa=sqrt(2*sstraina**2)
            end if
c     Calculate displacement for total decohesion
            nstrainf=2*Gic/nstressa
            sstrainf=2*Giic/sstressa
c     Calculate effective decohesion displacement
            if (strainNew(1).gt.0) then
              strainEfff=2*(Gic+(Giic-Gic)*(beta**2/(1+beta**2))**eta)/
     1        (KK*strainEffa)
            else
              strainEfff=sqrt(2*sstrainf**2)
            end if
c     Calculate damage variable
            d=strainEfff*(alpha-strainEffa)/(alpha*(strainEfff-strainEffa))
c     Calculate Ic and II
            do j=1,3
              do k=1,3
                Ic(j,k)=0
                if (j.eq.k) then
                  II(j,k)=1
                else
                  II(j,k)=0
                end if
              end do
            end do
            if (strainNew(1).lt.0) then
              Ic(1,1)=1
```

```
        end if
c     Calculate constitutive matrix
        do j=1,3
          do k=1,3
            DD(j,k)=0
          end do
        end do
        if (alpha.gt.strainEfff.or.stateOld(i,5).eq.1) then
          DD(1,1)=KK*Ic(1,1)
          stateNew(i,5)=1
        else if (alpha.lt.strainEffa) then
          do j=1,3
            DD(j,j)=KK
          end do
        else
          do j=1,3
            DD(j,j)=(1-d)*KK
          end do
          DD(1,1)=DD(1,1)+d*KK*Ic(1,1)
        end if
c     Update stresses
        do j=1,3
          stressNew(i,j)=0
        end do
        do j=1,3
          do k=1,3
            stressNew(i,j)=stressNew(i,j)+DD(j,k)*strainNew(k)
          end do
        end do
```

```
c
      end do
      end
```

A.2 Modified Linear Softening Model

```
      subroutine vumat(
c       Read Only Variables
     1  nblock,ndir,nshr,nstatev,nfieldv,nprops,lanneal,
     2  stepTime,totalTime,dt,cmname,coordMp,charLength,
     3  props,density,strainInc,relSpinInc,
     4  tempOld,stretchOld,defgradOld,fieldOld,
     5  stressOld,stateOld,enerInternOld,enerInelasOld,
     6  tempNew,stretchNew,defgradNew,fieldNew,
c       Write Only Variables
     7  stressNew,stateNew,enerInternNew,enerInelasNew)
c
      include 'vaba_param.inc'
c
      dimension props(nprops),density(nblock),coordMp(nblock),
     1  charLength(nblock),strainInc(nblock,ndir+nshr),
     2  relSpinInc(nblock,nshr),tempOld(nblock),
     3  stretchOld(nblock,ndir+nshr),defgradOld(nblock,ndir+nshr+nshr),
     4  fieldOld(nblock,nfieldv),stressOld(nblock,ndir+nshr),
     5  stateOld(nblock,nstatev),enerInternOld(nblock),
     6  enerInelasOld(nblock),tempNew(nblock),
     7  stretchNew(nblock,ndir+nshr),defgradNew(nblock,ndir+nshr+nshr),
     8  fieldNew(nblock,nfieldv),stressNew(nblock,ndir+nshr),
     9  stateNew(nblock,nstatev),enerInternNew(nblock),
```

```
     1  enerInelasNew(nblock)
c
      character*80 cmname
c
      real nstressa,sstressa,KK,Gic,Giic,ratio
      real alpha,beta,strainEfff,d,ef,e0,eta
      integer i,j,k,l
      real stressAbs,strainAbs,stressaEff
      real strainNew(ndir+nshr),strainEff,strainEffa
      real rshear
      real DD(3,3),Ic(3,3),II(3,3)
c     Separation ratio, decohesion displacement, fracture energy,
c         B-K parameter are defined here
      parameter (ratio=200,ef=0.02,Gic=0.969,Giic=1.719,eta=2.284)
c     Calculate propagation onset displacement
      e0=ef/ratio
c     Calculate displacement for propagation onset
      nstressa=2*Gic/ef
      sstressa=2*Giic/ef
      do i=1,nblock
c     Calculate <normal strength>
        if (stressOld(i,1).gt.0) then
          stressAbs=stressOld(i,1)
        else
          stressAbs=0
        end if
c     Calculate new displacement
        do j=1,ndir+nshr
          strainNew(j)=stateOld(i,j+1)+strainInc(i,j)
```

```
          stateNew(i,j+1)=strainNew(j)
        end do
c     Calculate <normal displacement>
        if (strainNew(1).gt.0) then
          strainAbs=strainNew(1)
        else
          strainAbs=0
        end if
c     Calculate effective displacement and shear displacement
        strainEff=strainAbs**2
        rshear=0
        do j=1,nshr
          rshear=rshear+4*strainNew(ndir+j)**2
        end do
        strainEff=strainEff+rshear
        strainEff=sqrt(strainEff)
        rshear=sqrt(rshear)
c     Determine maximum displacement reached in history
        alpha=stateOld(i,1)
        if (strainEff.gt.alpha) then
          alpha=strainEff
        end if
        stateNew(i,1)=alpha
c     Calculate displacement mode mixity
        beta=rshear/strainNew(1)
c     Calculate effective propagation onset displacement
        strainEffa=e0
c     Calculate effective decohesion displacement
        strainEfff=ef
```

```
c     Calculate effective strength
        if (strainNew(1).gt.0) then
          stressaEff=2*(Gic+(Giic-Gic)*
     1      (beta**2/(1+beta**2))**eta)/strainEfff
        else
          stressaEff=sstressa
        end if
c     Calculate effective penalty stiffness
        KK=stressaEff/e0
c     Calculate damage variable
        d=strainEfff*(alpha-strainEffa)/(alpha*(strainEfff-strainEffa))
c     Calculate Ic and II
        do j=1,3
          do k=1,3
            Ic(j,k)=0
            if (j.eq.k) then
              II(j,k)=1
            else
              II(j,k)=0
            end if
          end do
        end do
        if (strainNew(1).lt.0) then
          Ic(1,1)=1
        end if
c     Calculate constitutive matrix
        do j=1,3
          do k=1,3
            DD(j,k)=0
```

```
          end do
        end do
        if (alpha.gt.strainEfff.or.stateOld(i,5).eq.1) then
          DD(1,1)=KK*Ic(1,1)
          stateNew(i,5)=1
        else if (alpha.lt.strainEffa) then
          do j=1,3
            DD(j,j)=KK
          end do
        else
          do j=1,3
            DD(j,j)=(1-d)*KK
          end do
          DD(1,1)=DD(1,1)+d*KK*Ic(1,1)
        end if
c     Update stresses
        do j=1,3
          stressNew(i,j)=0
        end do
        do j=1,3
          do k=1,3
            stressNew(i,j)=stressNew(i,j)+DD(j,k)*strainNew(k)
          end do
        end do
c
      end do
      end
```

A.3 Modified Power-law Softening Model

```
      subroutine vumat(
c       Read Only Variables
     1  nblock,ndir,nshr,nstatev,nfieldv,nprops,lanneal,
     2  stepTime,totalTime,dt,cmname,coordMp,charLength,
     3  props,density,strainInc,relSpinInc,
     4  tempOld,stretchOld,defgradOld,fieldOld,
     5  stressOld,stateOld,enerInternOld,enerInelasOld,
     6  tempNew,stretchNew,defgradNew,fieldNew,
c       Write Only Variables
     7  stressNew,stateNew,enerInternNew,enerInelasNew)
c
      include 'vaba_param.inc'
c
      dimension props(nprops),density(nblock),coordMp(nblock),
     1  charLength(nblock),strainInc(nblock,ndir+nshr),
     2  relSpinInc(nblock,nshr),tempOld(nblock),
     3  stretchOld(nblock,ndir+nshr),defgradOld(nblock,ndir+nshr+nshr),
     4  fieldOld(nblock,nfieldv),stressOld(nblock,ndir+nshr),
     5  stateOld(nblock,nstatev),enerInternOld(nblock),
     6  enerInelasOld(nblock),tempNew(nblock),
     7  stretchNew(nblock,ndir+nshr),defgradNew(nblock,ndir+nshr+nshr),
     8  fieldNew(nblock,nfieldv),stressNew(nblock,ndir+nshr),
     9  stateNew(nblock,nstatev),enerInternNew(nblock),
     1  enerInelasNew(nblock)
c
      character*80 cmname
c
```

```
      real nstressa,sstressa,Gic,Giic,ratio,a1,a2,Gc,b1,b2,b3,d
      real alpha,beta,strainEfff,eta,ef,e0
      integer i,j,k,l
      integer F
      real stressAbs,strainAbs
      real strainNew(ndir+nshr),strainEff,strainEffa
      real rshear,DD(3,3)
c     Separation ratio, decohesion displacement, fracture energy,
c         B-K parameter are defined here
      parameter (ratio=200,ef=0.02,Gic=0.969,Giic=1.719,eta=2.284)
c     Calculate propagation onset displacement
      e0=ef/ratio
c     Calculate displacement for propagation onset
      nstressa=2*Gic/ef
      sstressa=2*Giic/ef
      do i=1,nblock
c     Calculate <normal strength>
        if (stressOld(i,1).gt.0) then
          stressAbs=stressOld(i,1)
        else
          stressAbs=0
        end if
c     Calculate new displacement
        do j=1,ndir+nshr
          strainNew(j)=stateOld(i,j+1)+strainInc(i,j)
          stateNew(i,j+1)=strainNew(j)
        end do
c     Calculate <normal displacement>
        if (strainNew(1).gt.0) then
```

```
          strainAbs=strainNew(1)
        else
          strainAbs=0
        end if
c     Calculate effective displacement and shear displacement
        strainEff=strainAbs**2
        rshear=0
        do j=1,nshr
          rshear=rshear+4*strainNew(ndir+j)**2
        end do
        strainEff=strainEff+rshear
        strainEff=sqrt(strainEff)
        rshear=sqrt(rshear)
c     Determine maximum displacement reached in history
        alpha=stateOld(i,1)
        if (strainEff.gt.alpha) then
          alpha=strainEff
        end if
        stateNew(i,1)=alpha
c     Calculate loading function
        if (strainEff.gt.stateOld(i,1)) then
          F=1
        else
          F=0
        end if
c     Calculate displacement mode mixity
        beta=rshear/strainNew(1)
c     Calculate effective decohesion displacement
        strainEfff=ef
```

```
c     Calculate effective propagation onset displacement
        strainEffa=e0
c     Calculate effective strength
        if (strainNew(1).gt.0) then
          stressaEff=2*(Gic+(Giic-Gic)*
     1      (beta**2/(1+beta**2))**eta)/strainEfff
        else
          stressaEff=sstressa
        end if
c     Calculate effective energy
        Gc=stressaEff*ef/2
c     Calculate damage variable
        d=((ef-alpha)/(ef-e0))**2
c     Calculate variables used in constitutive relations
        a1=-2*Gc*(ratio**2)/(ef**3)
        a2=4*Gc*ratio/(ef**2)
        b1=stressaEff/((ef-e0)**2)
        b2=-2*stressaEff*ef/((ef-e0)**2)
        b3=stressaEff*(ef**2)/((ef-e0)**2)
c     Update Stresses
        do j=1,3
          stressNew(i,j)=0
        end do
        if (alpha.gt.strainEfff.or.stateOld(i,5).eq.1) then
          stateNew(i,5)=1
        else if (alpha.lt.strainEffa) then
          do j=1,3
            if (strainNew(j).gt.0) then
              stressNew(i,j)=a1*(strainNew(j)**2)+a2*strainNew(j)
```

```
            else
              stressNew(i,j)=-a1*(strainNew(j)**2)+a2*strainNew(j)
            end if
          end do
        else
          if (F.eq.1) then
            do j=1,3
              if (strainNew(j).gt.0) then
                stressNew(i,j)=b1*(strainNew(j)**2)+b2*strainNew(j)+b3
              else
                stressNew(i,j)=-b1*(strainNew(j)**2)+b2*strainNew(j)-b3
              end if
            end do
          else
            stressNew(i,1)=d*nstressa*strainNew(1)/alpha
            stressNew(i,2)=d*sstressa*strainNew(2)/alpha
            stressNew(i,3)=d*sstressa*strainNew(3)/alpha
          end if
        end if
        if (strainNew(1).lt.0) then
          stressNew(i,1)=a2*strainNew(1)
        end if
c
      end do
      end
```