



Graduate Theses, Dissertations, and Problem Reports

2016

Determination of material properties for progressive damage analysis in carbon epoxy laminates using Abaqus

Javier Cabrera Barbero

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Cabrera Barbero, Javier, "Determination of material properties for progressive damage analysis in carbon epoxy laminates using Abaqus" (2016). *Graduate Theses, Dissertations, and Problem Reports*. 5296. <https://researchrepository.wvu.edu/etd/5296>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

**DETERMINATION OF MATERIAL PROPERTIES FOR
PROGRESSIVE DAMAGE ANALYSIS IN CARBON
EPOXY LAMINATES USING ABAQUS**

JAVIER CABRERA BARBERO



WEST VIRGINIA UNIVERSITY

Thesis submitted to the Benjamin M. Statler College of Engineering and Mineral Resources at
West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

Ever J. Barbero, Ph.D.

Bruce S. Kang, Ph.D.

Eduardo M. Sosa, Ph.D.

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

Morgantown, West Virginia

Keywords: Composite Materials, Damage, Cracks, Abaqus, Progressive Damage
Analysis, Discrete Damage Mechanics model.

Copyright 2016 Javier Cabrera Barbero

Abstract

Well-designed laminated composites do not fail suddenly but rather develop microscopic progressive damage that leads to changes in macroscopic material response, such as matrix cracks, stiffness reduction, and failure. Simulation techniques are able to predict damage initiation and evolution as a function of service conditions. A method for obtaining material properties for damage analysis of Glass and Carbon fiber composites is proposed using progressive damage analysis (PDA) model implemented in Abaqus.

The predictive capability of Progressive Damage Analysis (PDA) methods relies on material properties that characterize the ability of the composite to resist damage initiation and to delay damage progression. Although elastic moduli data and standard experimental methods exist, data and methods do not exist for damage-related properties. However, experimental data displaying macroscopic effects of damage (e.g., crack density and stiffness reduction) exists for a number of material systems. These experimental methods are sufficiently standardized to be used for other material systems.

The purpose of this study is to develop a method to obtain the missing material properties by adjusting their values so that the predicted material response matches experimental data. This methodology is based on minimizing the error between simulation predictions and available experimental data. Once the material properties are obtained, the simulation predictions are compared to a broad set of experimental data. Finally, sensitivity and convergence of Abaqus PDA is also studied.

Contents

1	Introduction	1
1.0.1	Objective	2
2	Progressive Damage Analysis (PDA)	3
2.0.1	Damage initiation	3
2.0.2	Damage evolution	5
3	Discrete Damage Mechanics (DDM)	8
3.1	Plate kinematics	9
3.2	Shear Lag Equations in Matrix Form	11
3.3	Solution of the Equilibrium Equation	12
3.4	Boundary Conditions for $\Delta T = 0$	13
3.4.1	(a) Stress-free at the Cracks Surfaces	14
3.5	Boundary Conditions for $\Delta T \neq 0$	14
3.6	Degraded Laminate Stiffness and CTE	15
3.7	Degraded Lamina Stiffness	16
3.8	Degraded Lamina Stiffness	17
3.9	Solution Algorithm	19
3.9.1	Lamina Iterations	19
3.9.2	Laminate Iterations	20
4	Methodology	21
4.1	Abaqus Script	21
4.2	Determination of material properties	22
4.3	Optimization problem	23
4.3.1	MATLAB® script controlling Abaqus	23
4.3.2	Abaqus® script	25
4.4	Generate Modulus Reduction Data	26
4.4.1	IM6/Avimid ®K Polymer	29
4.4.2	T300/Fiberite 934	30
4.4.3	AS4/Hercules 3501-6	32
4.4.4	IM7/MTM45-1	32

5	Results and Adjusted Parameters	34
5.1	E-glass epoxy laminate	34
5.2	Carbon epoxy laminate	36
5.3	In-situ strength	38
5.4	Convergence and Mesh Sensitivity	39
6	Conclusions	42
A	Abaqus's Python script: example	44
	Bibliography	50

List of Figures

1.1	Matrix cracks in the ± 70 laminas of a $[0/\pm 70_4/0_{1/2}]_s$ laminate loaded to 0.7% tensile strain along the 0° [1–5].	2
2.1	Energy dissipation property for one damage mode	7
3.1	Lamina and Laminate coordinate systems.	9
3.2	Representative unit cell.	9
4.1	Force vs. displacements for a laminate with loss of stiffness.	23
4.2	Convergence of the error vs. the number of iterations with <i>fminsearch</i> function..	24
4.3	Convergence of the error vs. the number of iterations with <i>fmin</i> function.	26
4.4	A comparison of Normalized Young Modulus between glass and carbon laminate with same LSS: $[0_2/90_4]_s$	27
4.5	DDM model prediction and crack density data in $[0/90_3]_s$ laminate IM6/Avimid®K Polymer.	30
4.6	Modulus reduction generated by DDM for several material systems: $[0/90_3]_s$ laminate IM6/Avimid®K Polymer, $[0/90_4]_s$ laminate T300/Fiberite 934, $[0/90_2]_s$ laminate AS4/Hercules 3501-6 and $[0/90_4]_s$ laminate IM7/MTM45-1.	30
4.7	DDM model prediction and crack density data in $[0_2/90_4]_s$ laminate T300/Fiberite 934.	31
4.8	DDM model prediction and crack density data in $[0/90_2]_s$ laminate AS4/Hercules 3501-6.	32
4.9	DDM model prediction and crack density data in $[0/90_4]_s$ laminate IM7/MTM45-1.	33
4.10	DDM model prediction and crack density data in $[0/\pm 55_4/0_{1/2}]_s$ laminate IM7/MTM45-1.	33
5.1	Normalized modulus vs. applied strain for laminate $[0/90_4]_s$ Fiberite/HyE-9082A.	35
5.2	Normalized modulus vs. applied strain for laminate $[0/\pm 40_4/0_{1/2}]_s$ Fiberite/HyE-9082A.	35
5.3	PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_4]_s$ IM7/MTM45-1.	36

5.4	PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode II for $[0/\pm 55_4/0_{1/2}]_s$ IM7/MTM45-1.	36
5.5	PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_3]_s$ IM6/Avimid®K Polymer.	37
5.6	PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_4]_s$ T300/Fiberite 934.	38
5.7	PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_2]_s$ AS4/Hercules 3501.	38
5.8	Non-dimensional strength predictions vs. ply thickness for transverse tensile strength.	39
5.9	Force vs applied strain for laminate 12 as function of number of elements.	40
5.10	PDA model prediction vs. longitudinal modulus from $[0/90_4]_s$ laminate as function of number of elements.	40
5.11	PDA model prediction vs longitudinal modulus from laminate $[0/90_4]_s$ IM7/MTM45-1 as function of type of element.	41

List of tables

4.1	Unidirectional ply properties for laminates IM6/Avimid K, T300/Fiberite 934, AS4/Hercules 3501-6 and IM7/MTM45-1 [1–4].	29
4.2	Laminates considered in this study.	31
4.3	Critical energy release rates for Discrete Damage Mechanics (<i>DDM</i>) model [5].	31
5.1	PDA optimal parameters calibrated with different elements.	34
5.2	In-situ strengths and PDA fracture energy G_{mt}^c	37
5.3	Comparison of adjusted and calculated in-situ strength values.	39

Chapter 1

Introduction

Composite materials are heterogeneous by definition. This anisotropy complicates the design and analysis of composite structures. At the macroscopic level, it is useful to forget about the microscopic details of the material and treat the composite as a homogeneous material. This is possible by homogenizing the properties of the constituent materials to come up with equivalent properties for the composite. This is achieved through of the micromechanics, which is the field that studies how to predict the properties of a composite. Other properties need further refinement to be accurately predicted by the micromechanics models.

The necessity of models that can predict properties based on the geometry and character of the constitutive materials is evident if we consider the anisotropy of composite (the value of the properties change with the orientation and significantly with the geometry and relative amounts of the constituent materials). It would be very costly to explore these two aspects only with experiments and thus, it is critical the developed of analytical and numerical models to predict the composite behavior. The most popular constitutive model is given by the Classical Laminate Theory (CLT). This theory, is useful to relate the strains with loads once it is given the stiffness and coefficient of thermal expansion (CTE) of each lamina.

An important property useful to the engineer is the strength of the composite but it is a difficult property to predict due to the nature of the events that lead to sudden failure and amount of defects present in the manufacturing process (fiber alignment, voids, residual stresses) since they play a significant role in the capacity of the composite. Different modes of failure can be recognized during the tests in composites.

Sometimes the failure mode is not critical to the structural stability or does not compromise the structure, namely, the composite degrades but is still capable of support the loads. This oftentimes happen with matrix cracking which is typically the first kind of damage that appears within the laminate. After that, the intralaminar cracking propagates stable or unstable generating a new modes of failure as delamination due to interlaminar cracks, interface cracks

between fiber and matrix or fiber breakage due to stress concentrations points. An example of the matrix cracking is shown in Figure 1.1 for a $[0/\pm 70_4/0_{1/2}]_s$ E-glass epoxy laminate. The cracks appear in the ± 70 laminae since the majority of the load in that direction is supported by the matrix while in 0° direction the load is supported by the fibers that are stiffer.

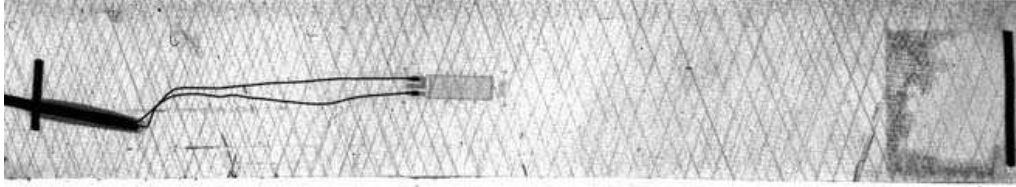


Figure 1.1: Matrix cracks in the ± 70 laminae of a $[0/\pm 70_4/0_{1/2}]_s$ laminate loaded to 0.7% tensile strain along the 0° [1–5].

The earliest, simplest and least accurate modeling technique to address matrix damage is perhaps the ply discount method [6, 1, Section 7.3.1]. Although many other models exist [7–15], etc., and several plugins are available [16, 17], this manuscript works focuses on the PDA model available in Abaqus. This is because of the broad availability of Abaqus, while plugins are not free, and the remaining models available in the literature are for the most part not readily available to be used in conjunction with commercial FEA environments.

1.0.1 Objective

The purpose of this study is to develop a methodology to obtain the missing material properties used in Abaqus by adjusting their values so that the predicted material response matches experimental data. The model used in Abaqus is so called Progressive Damage Analysis (PDA) explained in Chapter 2. This methodology is based on minimizing the error between simulation predictions and available experimental data developed in Chapter 4. For Carbon/polymer composites, a more elaborate method is developed because of lack of experimental stiffness reduction data in the literature. First, we calculate stiffness reduction data using an intermediate discrete damage mechanics model using experimental crack density data available in the literature. Then, the calculated stiffness reduction is used to obtain the properties of PDA in the same way as for Glass/polymer composites. The DDM model is explained in detail on Chapter 3. Once the material properties are obtained, the simulation predictions are compared to a broad set of experimental data shown in Chapter 5 including, sensitivity and convergence of PDA in Abaqus. Finally, conclusions are presented in Chapter 6.

Chapter 2

Progressive Damage Analysis (PDA)

The Abaqus PDA model is a generalization of an interlaminar decohesion model [18,19]. It assumes linearly elastic behavior of the undamaged material and it is used in combination with Hashin's damage initiation criteria [20,21].

2.0.1 Damage initiation

Abaqus assumes linear elastic behavior of the undamaged material. The Hashin's criterion can be defined for each mode (fiber tension, fiber compression, matrix tension, matrix compression) and defines the onset of material damage. Such four different damage initiation mechanisms, which can be coupled or uncoupled, are defined as follows

Fiber tension ($\sigma_{11} \geq 0$)

$$I_f^t = \left(\frac{\sigma_{11}}{F_{1t}} \right)^2 + \alpha \left(\frac{\sigma_{12}}{F_6} \right)^2 \quad (2.1)$$

Fiber compression ($\sigma_{11} < 0$)

$$I_f^c = \left(\frac{\sigma_{11}}{F_{1c}} \right)^2 \quad (2.2)$$

Matrix tension and/or shear ($\sigma_{22} \geq 0$)

$$I_m^t = \left(\frac{\sigma_{22}}{F_{2t}} \right)^2 + \left(\frac{\sigma_{12}}{F_6} \right)^2 \quad (2.3)$$

Matrix compression ($\sigma_{22} < 0$)

$$I_m^c = \left(\frac{\sigma_{22}}{2F_4} \right)^2 + \left[\left(\frac{F_{2c}}{2F_4} \right)^2 - 1 \right] \frac{\sigma_{22}}{F_{2c}} + \left(\frac{\sigma_{12}}{F_6} \right)^2 \quad (2.4)$$

where σ_{ij} are the components of the stress tensor; F_{1t} and F_{1c} are the tensile and compressive strengths in the fiber direction; F_{2t} and F_{2c} ; are the tensile and compressive strengths in the transverse direction; F_6 and F_4 are the longitudinal and transverse shear strengths, and α is determines the contribution of shear stress to the fiber tension mode. To obtain the model proposed by Hashin and Rotem [20] we set $\alpha = 0$ and $F_4 = (1/2)F_{2c}$. Furthermore, I_{ft} , I_{fc} ,

I_{mt} and I_{mc} are the failure indexes that indicate whether a damage initiation criterion has been satisfied for any damage mode. The onset of damage occurs when any of the indexes exceeds the value 1.0. Note that the strength values are material properties that must be provided by the user. Due to differences between testing of unidirectional composite and application conditions, the strength values measured by standard methods do not match accurately with the onset of damage. For accurate prediction of damage onset, the transverse tensile and shear strength values must be replaced by the so called *in-situ strength* of unidirectional lamina for transverse tensile strength (F_{2t}^{is}) and shear strength (F_6^{is}). In other words, once a lamina is embedded in a laminate, it takes more than the unidirectional strength to break it but rather the in-situ strength is required [6, section 7.4]. However, FEA commercial codes do not distinguish between in-situ and nominal strength values. There are two alternatives to obtain the in-situ values. One is to calculate them in terms of the unidirectional ply strength ply thickness [6, section 7.4]. The other is to adjust the values using laminate experimental data as proposed in this work.

Once damage starts, the effect of damage is taken into account by updating the values of stiffness coefficients [21] as follows

$$\sigma = C : \epsilon \quad (2.5)$$

where σ is the apparent stress, ϵ the strain, and C the damaged stiffness matrix. In addition, the damaged stiffness matrix is given by

$$C = \begin{pmatrix} (1 - d_f)E_1/\Delta & (1 - d_f)(1 - d_m)\nu_{21}E_1/\Delta & 0 \\ (1 - d_f)(1 - d_m)\nu_{12}E_2/\Delta & (1 - d_m)E_2/\Delta & 0 \\ 0 & 0 & (1 - d_s)G_{12} \end{pmatrix} \quad (2.6)$$

$$\Delta = 1 - (1 - d_f)(1 - d_m)\nu_{12}\nu_{21}$$

$$d_s = 1 - (1 - d_f^t)(1 - d_f^c)(1 - d_m^t)(1 - d_m^c)$$

where E_1 and E_2 are the moduli in fiber and matrix direction, G_{12} is the in-plane shear modulus, ν_{12} and ν_{21} are the Poisson's ratios, and d_f^t , d_f^c , d_m^t , d_m^c and d_s are the damage variables for fiber, matrix, and shear damage modes in tension and compression respectively. Note that the shear damage variable d_s is not independent, namely it depends of the remaining damage variables.

The damage variables for fiber and matrix in tension and compression, d_f^t , d_f^c , d_m^t and d_m^c , correspond to the four damage initiation modes given by equations (2.1-2.4). At any instant of time, each variable is updated according whether it is in tension or compression as follows

$$d_f = \begin{cases} d_f^t & \text{if } \sigma_{11} \geq 0 \\ d_f^c & \text{if } \sigma_{11} < 0 \end{cases} \quad (2.7)$$

and

$$d_m = \begin{cases} d_m^t & \text{if } \sigma_{22} \geq 0 \\ d_m^c & \text{if } \sigma_{22} < 0 \end{cases} \quad (2.8)$$

Abaqus PDA uses the model proposed by Matzenmiller et al. [19] to compute the degradation of stiffness matrix coefficients. The equations (2.1-2.4) are then used as damage evolution criteria by introducing effective stresses (2.9) instead of nominal stress. The relationship between the effective stress $\tilde{\sigma}$ and the apparent stress σ is computed [1] as follows

$$\tilde{\sigma} = M^{-1} : \sigma \quad (2.9)$$

and the damage effect tensor in Voigt notation is given as

$$M^{-1} = \begin{pmatrix} (1 - d_f)^{-1} & 0 & 0 \\ 0 & (1 - d_m)^{-1} & 0 \\ 0 & 0 & (1 - d_s)^{-1} \end{pmatrix} \quad (2.10)$$

Prior to damage initiation, the damage effect tensor M^{-1} is equal to the identity matrix, so $\tilde{\sigma} = \sigma$. Once the damage has started for at least one mode, the damage effect tensor becomes significant in the criteria for damage initiation of other modes. When $d_f = 0$ and $d_m = d_s = 1$, equation (2.10) represents the ply discount method.

2.0.2 Damage evolution

Once any damage initiation criteria is satisfied, further loading will cause degradation of material stiffness. The evolution of the damage variable employs four critical energy dissipation G_i^c , which correspond to each damage mode: fiber tension ($i = ft$), fiber compression ($i = fc$), matrix tension ($i = mt$) and matrix compression ($i = mc$). So, in addition to six strength values, four critical energy dissipation properties must be provided. The triangle's area OAC shown in Figure 2.1 corresponds to this critical energy dissipation for each mode. Note that this contrasts with other models where the onset and evolution damage are predicted just in terms of the critical energy release rate (ERR) [22].

Normally, the constitutive model is expressed in terms of stress-strain, but when the material exhibits a strain-softening behavior, as shown in Figure 2.1 along line AC , such formulation produces strong mesh and element-type dependent results, while in reality the actual composite behaves the same regardless of what mesh or element is used. In order to alleviate the mesh dependency, PDA uses a so-called *characteristic length* (L_c) to transform the PDA constitutive model from *stress-strain* to *strain-displacement* by computing $\delta = \epsilon \cdot L_c$. Using the characteristic length relieves some but not all of the mesh dependency. Furthermore note that PDA does not resolve the actual cracks in the composite, the crack density is not calculated and only the reduction of stiffness can be calculated in terms of damage values that cannot be directly

measured experimentally but only inferred from stiffness reduction.

The evolution of damage variables is governed by an equivalent displacement δ^{eq} shown in Figure 2.1. In this way, each damage mode is represented as a 1D stress-displacement problem. The equivalent displacement for each mode is expressed in terms of the effective stress components used in the initiation criterion for each damage mode. Such 1D displacements and 1D stresses are defined as follows [21]:

Fiber tension ($\sigma_{11} \geq 0$)

$$\begin{aligned}\delta_{ft}^{eq} &= L^c \sqrt{\langle \epsilon_{11} \rangle^2 + \alpha \epsilon_{12}^2} \\ \sigma_{ft}^{eq} &= \frac{\langle \sigma_{11} \rangle \langle \epsilon_{11} \rangle + \alpha \sigma_{12} \epsilon_{12}}{\delta_{ft}^{eq} / L^c}\end{aligned}\quad (2.11)$$

Fiber compression ($\sigma_{11} < 0$)

$$\begin{aligned}\delta_{fc}^{eq} &= L^c \langle -\epsilon_{11} \rangle \\ \sigma_{fc}^{eq} &= \frac{\langle -\sigma_{11} \rangle \langle -\epsilon_{11} \rangle}{\delta_{fc}^{eq} / L^c}\end{aligned}\quad (2.12)$$

Matrix tension and/or shear ($\sigma_{22} \geq 0$)

$$\begin{aligned}\delta_{mt}^{eq} &= L^c \sqrt{\langle -\epsilon_{22} \rangle^2 + \epsilon_{12}^2} \\ \sigma_{mt}^{eq} &= \frac{\langle \sigma_{22} \rangle \langle \epsilon_{22} \rangle + \sigma_{12} \epsilon_{12}}{\delta_{mt}^{eq} / L^c}\end{aligned}\quad (2.13)$$

Matrix compression ($\sigma_{22} < 0$)

$$\begin{aligned}\delta_{mc}^{eq} &= L^c \sqrt{\langle -\epsilon_{22} \rangle^2 + \epsilon_{12}^2} \\ \sigma_{mc}^{eq} &= \frac{\langle -\sigma_{22} \rangle \langle -\epsilon_{22} \rangle + \sigma_{12} \epsilon_{12}}{\delta_{mc}^{eq} / L^c}\end{aligned}\quad (2.14)$$

where $\langle \cdot \rangle$ represents the Macaulay operator defined as $\langle \eta \rangle = \frac{1}{2} (\eta + |\eta|)$ for every $\eta \in \mathfrak{R}$.

For each mode, the damage variable varies from zero (undamaged) to one (totally damaged). The damage variable for a particular mode is derived using Figure 2.1 as follows

$$d = \frac{\delta_c^{eq} (\delta^{eq} - \delta_o^{eq})}{\delta^{eq} (\delta_c^{eq} - \delta_o^{eq})}\quad (2.15)$$

where δ_c^{eq} is the maximum value of δ^{eq} at point C in Fig 2.1, for each mode.

In PDA, a material point is initially stressed and strained along the linear elastic line OA in Fig 2.1, with a initial structural stiffness E/L_c given by the slope of line OA until the stress reaches the in-situ strength (point A). In-situ transverse tension and in-plane shear are larger than nominal strength values [6, section 7.2.1]. In-situ values can be obtained (adjusted) by

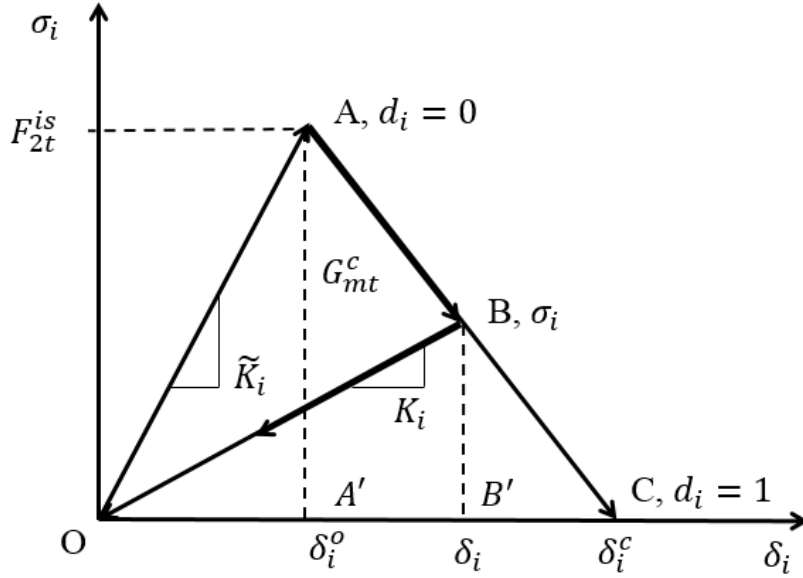


Figure 2.1: Energy dissipation property for one damage mode

matching PDA model predictions to initiation of laminate modulus reduction observed experimentally. One starts with the known unidirectional value as initial guess, and lets the optimization algorithm adjust the in-situ strength until a match is found.

After point A the material undergoes progressive damage and stress softening, i.e., due to damage, the value of stress goes down along line ABC . If the load is released at B , the material returns to the origin following an elastic-damaged path with reduced structural stiffness given by the slope of BO . The area OAC represents dissipated energy per unit crack area G_{mt}^c when the material is 100% degraded. No experimental method exists for measuring G_{mt}^c , but it can be evaluated indirectly by adjusting its value so that predicted laminate stiffness matches experimental values [5]. The critical energy release rate for interlaminar damage for a similar material system is used as initial guess for optimization.

Chapter 3

Discrete Damage Mechanics (DDM)

Discrete damage mechanics is based on the use of discrete fracture mechanics to predict damage initiation and evolution [22] for symmetric laminate under in-plane loads. Each lamina in the laminate is susceptible to matrix cracking which is controlled by a damage activation function $g(\lambda, \epsilon, \Delta T)$, discussed in detail in section []. Such function tells us whether the lamina suffers new cracks or not, namely it takes into account if the total energy obtained at this point is enough to produce a new crack. The energy stored is a function of the applied strain ϵ , the crack density λ and ΔT which is the difference between the reference temperature and the operation temperature.

The applied strain is incremented from zero up to a certain number in finite increments. In each load step, the damage activation function is calculated for each lamina. If $g(\lambda, \epsilon, \Delta T) < 0$, then no damage occurs but whether $g(\lambda, \epsilon, \Delta T) > 0$, a new crack/s are generated parallel to the fiber orientation. Once we know that a new crack appears, a return mapping algorithm is then used to determine the current crack density of such lamina and so on until all the laminas within the laminate have been analyzed and finally converged. At this point, the laminate does not suffer more matrix cracking until the load is further incremented and the procedure is repeated again. As the result of this increasing crack density, there is a degradation of elastic properties that can be calculated also lamina by lamina.

When a new crack is generated in one lamina, this produces a load drop which is redistributed into the remaining laminas causing a loss of laminate stiffness. In each lamina, the damage caused by the cracks is represented by the crack density, $\lambda = 1/(2l)$ and it is defined as the inverse of the distance between two consecutive cracks. The lamina coordinate system is denoted by x_1, x_2 and x_3 and the laminate coordinate system by x, y and z , as shown in Figure 3.1.

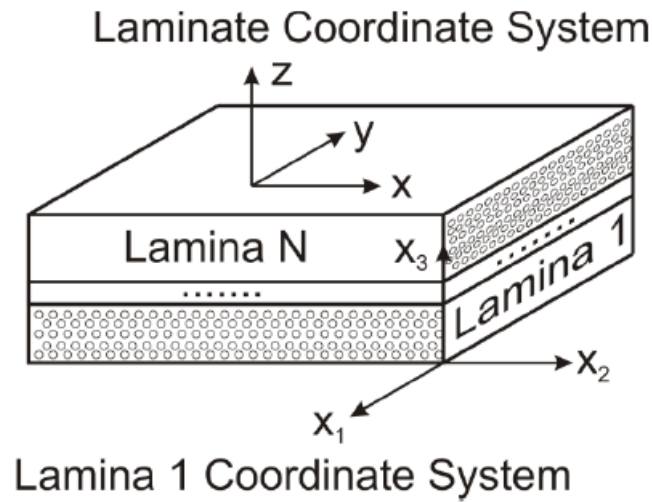


Figure 3.1: Lamina and Laminate coordinate systems.

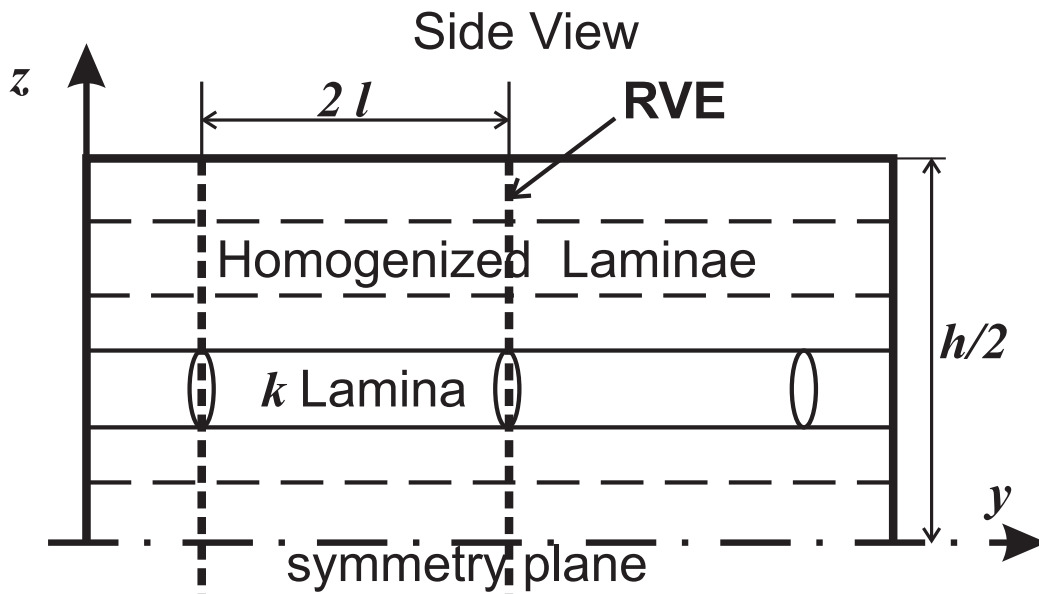


Figure 3.2: Representative unit cell.

The representative volume element (RVE) used in the analysis is delimited by the bottom-surface and the mid-surface (symmetric) of the laminate, a unit length in fiber direction, and the distance between two consecutive cracks ($2l$) = $1/\lambda$ as shown the Figure 3.2.

3.1 Plate kinematics

The most practical laminates are symmetric so that they are the most efficient to design structures loaded by membrane loads [6, Chapter 12]. Therefore, no bending moments are applied to the laminate

$$\frac{\partial w^i}{\partial x} = \frac{\partial w^i}{\partial y} = 0 \quad (3.1)$$

where the superscript (i) refers to the *i*th lamina.

The bottom and top surfaces of the laminate are stress-free and the laminate is thin enough to consider a plane stress state.

The thickness average of any mechanical variable is defined as

$$\hat{\phi} = \frac{1}{h_i} \int_{h_i} \phi \, dx_3 \quad (3.2)$$

where ϕ can be any parameter such as $\sigma, \epsilon, Q, \dots$

This definition is useful for obtaining the overall reduced stiffness properties based on the averaged displacements.

For a cracking lamina *k*, the constitutive equation is

$$\hat{\sigma}_i^k = \bar{Q}_{ij}^k (\hat{\epsilon}_j - \hat{\alpha}_j^k \Delta T) \quad (3.3)$$

where $\hat{\alpha}_j^k$ is the CTE of lamina (*k*),

$$\hat{\sigma}^k = \begin{Bmatrix} \hat{\sigma}_1^k \\ \hat{\sigma}_2^k \\ \hat{\sigma}_{12}^k \end{Bmatrix} \quad (3.4)$$

and

$$\hat{\epsilon}^k = \begin{Bmatrix} \hat{u}_{,1}^k \\ \hat{v}_{,2}^k \\ \hat{u}_{,2}^k + \hat{v}_{,1}^k \end{Bmatrix} \quad (3.5)$$

where the overline denotes undamaged quantities and ,1; ,2 represents partial derivatives as usual.

The constitutive equations for the remaining laminas ($m \neq k$) can be calculated by using equation (3.3) with the reduced stiffness matrix, Q_{ij}^m , written in terms of their previously calculated damage values D_2^m, D_6^m , defined later in Section 3.7, and rotated to the *k* coordinate

system using the usual transformation [6]

$$Q^m = [T(-\theta)] \begin{bmatrix} \tilde{Q}_{11}^m & (1 - D_2^m)\tilde{Q}_{22}^m & 0 \\ (1 - D_2^m)\tilde{Q}_{12}^m & (1 - D_2^m)\tilde{Q}_{22}^m & 0 \\ 0 & 0 & (1 - D_6^m)\tilde{Q}_{66}^m \end{bmatrix} [T(\theta)]^T \quad (3.6)$$

The damage values belong to a diagonal second order damage tensor defined in [23].

3.2 Shear Lag Equations in Matrix Form

The functional form of the intralaminar shear stresses is assumed to be:

$$\tau_{13}^i(x_3) = \tau_{13}^{i-1,i} + \left[\tau_{13}^{i,i+1}(x) - \tau_{13}^{i-1,i}(x) \right] \frac{x_3 - x_3^{i-1,i}}{h^i} \quad (3.7)$$

and

$$\tau_{23}^i(x_3) = \tau_{23}^{i-1,i} + \left[\tau_{23}^{i,i+1}(x) - \tau_{23}^{i-1,i}(x) \right] \frac{x_3 - x_3^{i-1,i}}{h^i} \quad (3.8)$$

That is a linear variation in the x_3 direction (see Figure 3.1). Where $\tau_{13}^{i-1,i}$ is the shear stress at the interface between the $i-1$ th and the i th lamina, and $x_3^{i-1,i}$ is the position of the interface between $i-1$ th and i th lamina. This assumption is common to several other analytical models and is called the Shear Lag assumption [24].

The shear lag equations are obtained from the constitutive equations for out-of-plane shear strains and stresses by means of weighted averages:

$$\begin{aligned} \begin{Bmatrix} \hat{u}^{(i)} - \hat{u}^{(i-1)} \\ \hat{v}^{(i)} - \hat{v}^{(i-1)} \end{Bmatrix} &= \frac{h_{(i-1)}}{6} \begin{bmatrix} S_{45} & S_{55} \\ S_{44} & S_{45} \end{bmatrix}^{(i-1)} \begin{Bmatrix} \tau_{23}^{i-2,i-1} \\ \tau_{13}^{i-2,i-1} \end{Bmatrix} \\ &+ \left[\frac{h_{(i-1)}}{3} \begin{bmatrix} S_{45} & S_{55} \\ S_{44} & S_{45} \end{bmatrix}^{(i-1)} + \frac{h_{(i)}}{3} \begin{bmatrix} S_{45} & S_{55} \\ S_{44} & S_{45} \end{bmatrix}^{(i)} \right] \begin{Bmatrix} \tau_{23}^{i-1,i} \\ \tau_{13}^{i-1,i} \end{Bmatrix} \\ &+ \frac{h_{(i)}}{6} \begin{bmatrix} S_{45} & S_{55} \\ S_{44} & S_{45} \end{bmatrix}^{(i)} \begin{Bmatrix} \tau_{23}^{i,i+1} \\ \tau_{13}^{i,i+1} \end{Bmatrix} \end{aligned} \quad (3.9)$$

From which the intralaminar shear stresses are obtained as:

$$\begin{aligned}
 \tau_{23}^{i,i+1} - \tau_{23}^{i-1,i} &= \sum_{j=1}^{n-1} \left[[H]^{-1}_{2i-1,2j-1} - [H]^{-1}_{2i-3,2j-1} \right] \left\{ \hat{u}^{(j+1)} - \hat{u}^{(j)} \right\} \\
 &\quad + \left[[H]^{-1}_{2i-1,2j} - [H]^{-1}_{2i-3,2j} \right] \left\{ \hat{v}^{(j+1)} - \hat{v}^{(j)} \right\} \\
 \tau_{13}^{i,i+1} - \tau_{13}^{i-1,i} &= \sum_{j=1}^{n-1} \left[[H]^{-1}_{2i,2j-1} - [H]^{-1}_{2i-2,2j-1} \right] \left\{ \hat{u}^{(j+1)} - \hat{u}^{(j)} \right\} \\
 &\quad + \left[[H]^{-1}_{2i,2j} - [H]^{-1}_{2i-2,2j} \right] \left\{ \hat{v}^{(j+1)} - \hat{v}^{(j)} \right\} \tag{3.10}
 \end{aligned}$$

in terms of the $2(N-1)$ by $2(N-1)$ coefficient matrix H , which is the assemblage of equation (3.9). These relationships are then used in the equilibrium equations (3.11) and (3.12) to substitute u and v for τ_{13} and τ_{23} .

3.3 Solution of the Equilibrium Equation

The equilibrium equations for each lamina can be stated as follows

$$\hat{\sigma}_{1,1}^{(i)} + \hat{\tau}_{12,2}^{(i)} + \left(\hat{\tau}_{13}^{i,i+1} - \hat{\tau}_{13}^{i-1,i} \right) / h_i = 0 \tag{3.11}$$

$$\hat{\tau}_{12,1}^{(i)} + \hat{\sigma}_{2,2}^{(i)} + \left(\hat{\tau}_{23}^{i,i+1} - \hat{\tau}_{23}^{i-1,i} \right) / h_i = 0 \tag{3.12}$$

Then, the solution to solve the PDE (Partial Differential Equations) obtained in (3.11) and (3.12) is proposed in the following form

$$\begin{aligned}
 \hat{u}^{(i)} &= a_i \sinh \lambda_e x_2 + a x_1 + b x_2 \\
 \hat{v}^{(i)} &= b_i \sinh \lambda_e x_2 + b x_1 + a^* x_2 \tag{3.13}
 \end{aligned}$$

where e is the eigenvalue numbers. The general solution can be written as

$$\begin{pmatrix} \hat{u}^{(1)} \\ \hat{u}^{(2)} \\ \cdot \\ \cdot \\ \hat{u}^{(n)} \\ \hat{v}^{(1)} \\ \hat{v}^{(2)} \\ \cdot \\ \cdot \\ \hat{v}^{(n)} \end{pmatrix} = \sum_{e=1}^{2N} A_e \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \\ b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{pmatrix} \sinh(\eta_e x_2) + \begin{pmatrix} a \\ a \\ \cdot \\ \cdot \\ a \\ b \\ b \\ \cdot \\ \cdot \\ b \end{pmatrix} x_1 + \begin{pmatrix} b \\ b \\ \cdot \\ \cdot \\ b \\ a^* \\ a^* \\ \cdot \\ \cdot \\ a^* \end{pmatrix} x_2 \quad (3.14)$$

Where A_e , a , b and a^* in the general solution (3.14) are the coefficients that need to be found to generate the particular solution for each set of boundary conditions [25].

The next step is to evaluate each term in (3.11) and (3.12) using (3.14). This leads to the next eigenvalue problem:

$$\begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix} \begin{Bmatrix} a_j \\ b_j \end{Bmatrix} + \eta^2 \begin{bmatrix} \zeta_{26} & \zeta_{22} \\ \zeta_{66} & \zeta_{26} \end{bmatrix} \begin{Bmatrix} a_j \\ b_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (3.15)$$

When (3.14) is plugged in (3.11) and (3.12), an Eigenvalue system is formed, with eigenvalues λ_e and the eigenvectors $\begin{Bmatrix} a \\ b \end{Bmatrix}$. This system yields $2N$ eigenvalues and $2N$ eigenvectors. The $2N - 2$ non trivial eigenvalues correspond to the hyperbolic sine solutions, while the two trivial eigenvalues correspond to the linear solutions.

3.4 Boundary Conditions for $\Delta T = 0$

First consider the case of mechanical loads and no thermal loads. To find the values of A_e , a , a^* , b , the following boundary conditions are enforced: (a) stress-free at the crack surfaces, (b) external loads, and (c) homogeneous displacements. The boundary conditions are then assembled into an algebraic system as follows

$$[B] \{A_e, a, a^*, b\}^T = \{F\} \quad (3.16)$$

where $[B]$ is the coefficient matrix of dimensions $2N + 1$ by $2N + 1$; $\{A_e, a, a^*, b\}^T$ represents the $2N + 1$ unknown coefficients, and $\{F\}$ is the RHS or force vector, also of dimension $2N + 1$.

3.4.1 (a) Stress-free at the Cracks Surfaces

The surfaces of the cracks are stress-free

$$\int_{-1/2}^{1/2} \hat{\sigma}_2^{(k)}(x_1, l) dx_1 = 0 \quad (3.17)$$

$$\int_{-1/2}^{1/2} \hat{\tau}_{12}^{(k)}(x_1, l) dx_1 = 0 \quad (3.18)$$

3.4.1.1 (b) External Loads

In the direction parallel to the surface of the cracks (fiber direction x_1) the load is supported by all the laminas

$$\frac{1}{2l} \sum_{i=1}^N h_i \int_{-l}^l \hat{\sigma}_1^{(i)}(1/2, x_2) dx_2 = h \hat{\sigma}_1 \quad (3.19)$$

In the direction normal to the crack surface (x_2 direction) only the uncracking (homogenized) laminas carry load

$$\sum_{m \neq k} h_m \int_{1/2}^{1/2} \hat{\sigma}_2^{(m)}(x_1, l) dx_1 = h \hat{\sigma}_2 \quad (3.20)$$

$$\sum_{m \neq k} h_m \int_{1/2}^{1/2} \hat{\tau}_{12}^{(m)}(x_1, l) dx_1 = h \hat{\tau}_{12} \quad (3.21)$$

3.4.1.2 (c) Homogeneous Displacements

For a homogenized symmetric laminate, membrane loads produce a uniform displacement field through the thickness, i.e., all the uncracking laminas are subjected to the same displacement

$$\hat{u}^{(m)}(x_1, l) = \hat{u}^{(r)}(x_1, l) \quad ; \quad \forall m \neq k \quad (3.22)$$

$$\hat{v}^{(m)}(x_1, l) = \hat{v}^{(r)}(x_1, l) \quad ; \quad \forall m \neq k \quad (3.23)$$

where r is an uncracked lamina taken as reference. In the computer implementation, lamina 1 is taken as reference unless lamina 1 is cracking, in which case lamina 2 is taken as reference.

3.5 Boundary Conditions for $\Delta T \neq 0$

Next, consider the case of thermal loads, which add a constant term to the boundary conditions. Constant terms do not affect the matrix $[B]$, but rather subtract from the forcing vector $\{F\}$,

as follows

$$\{F\}_{\Delta T \neq 0} = \left\{ \begin{array}{c} \Delta T \sum_{j=1,2,6} \bar{Q}_{1j}^{(k)} \bar{\alpha}_j^{(k)} \\ \Delta T \sum_{j=1,2,6} \bar{Q}_{1j}^{(k)} \bar{\alpha}_j^{(k)} \\ \Delta T \sum_{i \neq (k)} \sum_{j=1,2,6} \bar{Q}_{1j}^{(i)} \bar{\alpha}_j^{(i)} \\ \Delta T \sum_{i \neq k} \sum_{j=1,2,6} \bar{Q}_{2j}^{(i)} \bar{\alpha}_j^{(i)} \\ \Delta T \sum_{i \neq k} \sum_{j=1,2,6} \bar{Q}_{6j}^{(i)} \bar{\alpha}_j^{(i)} \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \\ 0 \end{array} \right\} \quad (3.24)$$

In this way, the strain calculated for a unit thermal load ($\Delta T = 1$) is the degraded CTE of the laminate for the current crack density set λ .

3.6 Degraded Laminate Stiffness and CTE

First, we calculate the degraded stiffness of the laminate $Q = \mathbf{A}/h$ for a given crack density λ_k in a cracked lamina k , where \mathbf{A} is the in-plane laminate stiffness matrix, and h is the thickness of the laminate. First, the thickness-averaged strain field in all laminas can be obtained by using the equation (3.14). At this point, the homogenization problem replaces the cracks from the *RVE* by a reduction of stiffness of the homogenized material. Taking the volume average of the *RVE* as follows

$$\bar{\phi} = \int_V \phi dv \quad (3.25)$$

the constitutive equations are expressed in terms of stress and strain averaged. Since the CDM principle states that the applied strain is equal to the average strain at one point far enough where the cavity of cracks or inclusions are neglected, namely the *RVE*, the elastic constitutive equation is simplified as

$$\hat{\epsilon} = S \sigma_j^o \quad (3.26)$$

where σ_j^o is the stress applied to the laminate.

Then, the compliance of the laminate S in the coordinate system of lamina k can be calculated one column at a time by solving for the strains (3.14) for three load cases, a , b , and c , all with

$\Delta T = 0$, as follows

$$\sigma_a^o = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}; \quad \sigma_b^o = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}; \quad \sigma_c^o = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}; \quad \Delta T = 0 \quad (3.27)$$

Then, the compliance matrix of the laminate in the lamina k coordinate system as a function of the crack density is assembled as

$$S(\lambda_{matrix}) = \begin{bmatrix} a\epsilon_1 & b\epsilon_1 & c\epsilon_1 \\ a\epsilon_1 & b\epsilon_1 & c\epsilon_1 \\ a\gamma_{12} & b\gamma_{12} & c\gamma_{12} \end{bmatrix} \quad (3.28)$$

To get the degraded CTE of the laminate, one sets $\hat{\sigma}^o = \{0, 0, 0\}^T$ and $\Delta T = 1$. The resulting strain is equal to the CTE of the laminate, i.e., $\{\alpha_x, \alpha_y, \alpha_{xy}\}^T = \{\epsilon_1, \epsilon_2, \gamma_{12}\}^T$.

3.7 Degraded Lamina Stiffness

The stiffness of lamina m , with $m \neq k$, in the coordinate system of lamina k (see Figure []) is given by (3.6) in terms of the previously calculated values $D_2^{(m)}, D_6^{(m)}$, given by (3.31). The stiffness of the cracking lamina $Q^{(k)}$ is yet unknown. Note that all quantities are expressed in the coordinate system of lamina k .

The laminate stiffness is defined by the contribution of the cracking lamina k plus the contribution of the remaining $N - 1$ laminas, as follows

$$Q = Q^{(k)} \frac{h_k}{h} + \sum_{m=1}^n (1 - \delta_{mk}) Q^{(m)} \frac{h_m}{h} \quad (3.29)$$

where the *delta* Dirac is defined as $\delta_{mk} = 1$ if $m = k$, otherwise 0. The left-hand side (LHS) of (3.29) is known from (3.28) and all values of $Q^{(m)}$ can be easily calculated so that the m laminas are not cracking at the moment. Therefore, one can calculate the degraded stiffness $Q^{(k)}$ of lamina k as follows

$$Q^{(k)} = \frac{h}{h_k} \left[Q - \sum_{m=1}^n (1 - \delta_{mk}) Q^{(m)} \frac{h_m}{h} \right] \quad (3.30)$$

where Q without a superscript is the stiffness of the laminate. To facilitate later calculations, the stiffness $Q^{(k)}$ can be written in terms of the stiffness of the undamaged lamina and damage

variables $D_2^{(k)}, D_6^{(k)}$, using equation (3.6). The damage factor can be written as follows

$$D_j^{(k)}(\lambda_k, \epsilon^0) = 1 - Q_{jj}^{(k)} / \tilde{Q}_{jj}^{(k)} \quad ; \quad j = 2, 6; \text{ no sum on } j \quad (3.31)$$

where $\tilde{Q}^{(k)}$ is the original value of the undamaged property and $Q^{(k)}$ is the degraded (homogenized) value computed in (3.30), both expressed in the coordinate system of lamina k .

The coefficient of thermal expansion of the cracking lamina k is calculated in a similar fashion, as follows

$$\alpha^{(k)} = \frac{1}{h_k} S^{(k)} \left(h Q \alpha - \sum_{m \neq k} h_m Q^{(m)} \alpha^{(m)} \right) \quad (3.32)$$

with $S = [Q^{(k)}]^{-1}$. The corresponding thermal damage is calculated as

$$D_j^{\alpha(k)} = 1 - \alpha_j^{(k)} / \tilde{\alpha}_j^{(k)} \quad ; \quad j = 2, 6 \quad (3.33)$$

Once the damages for lamina k are known, they are used in the next laminate iteration as the material properties of the lamina, where it is homogenized attributing a loss of stiffness to the cracking lamina.

3.8 Degraded Lamina Stiffness

The DDM model [22] predicts when the crack density of a lamina should increase by means of a damage activation function $g(\lambda, \epsilon, \Delta T)$, which is essentially the Griffen fracture criteria, namely it is generated a new crack when

$$G(\lambda, \epsilon, \Delta T) \geq G_c \quad (3.34)$$

where G is the energy release rate (ERR) for the given laminate state $(\lambda, \epsilon, \Delta T)$ and G_c is the critical ERR that is a material property.

Since intralaminar cracks may propagate in mode I (opening) and mode II (shearing), the ERR needs to be decomposed into G_{Ic} and G_{IIc} . This is accomplished by evaluating all the equations in the coordinate system of the cracking lamina k . Then, G_I is calculated with $\epsilon = \{0, \epsilon_{22}, 0\}$ and G_{II} is calculated with $\epsilon = \{0, 0, \gamma_{12}, 0\}$ [26]. The proposed mode separation is consistent with the method of mechanical work during crack closure in classical fracture mechanics [27], which is the basis for the Virtual Crack Closure Technique (VCCT) broadly adopted in FEA.

The damage activation function may consider or not the interaction between Mode I and Mode II. If the damage activation function considers interaction, a proposed functional form is [28]

$$g(\lambda, \epsilon, \Delta T) = (1 - r) \sqrt{\frac{G_I(\lambda, \epsilon, \Delta T)}{G_{Ic}}} + r \frac{G_I(\lambda, \epsilon, \Delta T)}{G_{Ic}} G_{Ic} + \frac{G_{II}(\lambda, \epsilon, \Delta T)}{G_{IIc}} - 1 \leq 0 \quad (3.35)$$

where

$$r = \frac{G_{Ic}}{G_{IIc}} \quad (3.36)$$

The critical energy release rates and are not easily found in the literature and have to be fit to experimental data using a methodology that is explained in Chapter []. The energy release rates associated with the introduction of a new crack in the middle of the RVE can be calculated by computing the laminate stiffness and CTE for the current state and for a trial crack density that is the double current crack density. To find the energy associated to those states we use the Griffith's energy principle applied on its discrete (finite) form in order to describe the behavior of crack growth, as follows

$$\begin{aligned} G_I &= -\frac{\Delta U_I}{\Delta A} \\ G_{II} &= -\frac{\Delta U_{II}}{\Delta A} \end{aligned} \quad (3.37)$$

where $\Delta U_I, \Delta U_{II}$ are the change in laminate strain energy during mode I and mode II finite crack growth, respectively; and ΔA is the newly created (finite) crack area, which is one half of the new crack surface. Counting crack area as one-half of crack surface is consistent with the classical fracture mechanics convention for which fracture toughness G_c is twice of Griffith's surface energy γ_c .

To calculate the ERR, it is convenient to use the laminate stiffness Q in the c.s. of the cracked lamina, because in this way, the ERR can be decomposed into opening and shear modes. Since the laminate stiffness is available from the analysis as a function of crack density λ , the ERR can be calculated, for a fixed strain level (load), and using [26] [29, Section 3.2.10], into (3.37), we arrive at

$$G_I = -\frac{V}{2\Delta A} (\epsilon_2 - \alpha_2 \Delta T) \Delta Q_{2j} (\epsilon_j - \alpha_j \Delta T) \quad ; \quad \text{opening mode} \quad (3.38)$$

$$G_{II} = -\frac{V}{2\Delta A} (\epsilon_6 - \alpha_6 \Delta T) \Delta Q_{6j} (\epsilon_j - \alpha_j \Delta T) \quad ; \quad \text{shear mode} \quad (3.39)$$

where $V, \Delta A$ are the volume of the RVE and the increment of crack area, respectively; ΔQ_{ij} is

the change in laminate stiffness corresponding to the change in crack area; and all quantities are *laminate average* quantities expressed in the c.s of the cracked lamina in order to allow for ERR mode decomposition [26].

In the current implementation of the model, $\Delta A = 1/h_k$ is the area of one new crack appearing halfway between two existing cracks. In this case the crack density doubles and $\Delta Q = Q(2\lambda) - Q(\lambda) < 0$. Alternative crack propagation strategies are considered in [30]. It can be seen that the proposed methodology provides the key ingredients for the computation of the ERR; namely the degraded stiffness and degraded CTE of the laminate, both as a function of crack density.

The damage activation function (3.35) can now be calculated for any value of λ and applied strain $\epsilon_x, \epsilon_y, \gamma_{xy}$ applied to the laminate. Note that the computation of the ERR components derives directly from the displacement solution (3.14) for a discrete crack (Figure 3.2). When this formulation is used along with the finite element method (FEM), it does not display mesh dependency on the solution.

3.9 Solution Algorithm

The solution algorithm consists of (a) strain steps, (b) laminate-iterations, and (c) lamina-iterations. The state variables for the laminate are the array of crack densities for all laminas i and the membrane strain ϵ . At each load (strain) step, the strain on the laminate is increased and the laminas are checked for damage.

3.9.1 Lamina Iterations

When matrix cracking is detected in lamina k , a return mapping algorithm (RMA) is invoked to iterate and adjust the crack density λ_k in lamina k in such a way that g_k returns to zero while maintaining equilibrium between the external forces and the internal forces in the laminas. The iterative procedure works as follows. At a given strain level ϵ for the laminate and given λ_k for lamina k , calculate the value of the damage activation function g_k and the damage variables, which are both functions of λ_k . The RMA calculates the increment (decrement) of crack density as

$$\Delta\lambda_k = -g_k / \frac{\partial g_k}{\partial \lambda} \quad (3.40)$$

until $g_k = 0$ is satisfied within a given tolerance, for all $k = 1 \dots N$, where N is the number of laminas in the laminate. The analysis starts with a negligible value of crack density present in all laminas ($\lambda = 0.02$ cracks/mm were used in the examples) due to defects inherent into materials.

3.9.2 Laminate Iterations

To calculate the stiffness reduction of a cracked lamina (k -lamina), all of the other laminas (m -laminas) in the laminate are considered not damaging during the course of lamina-iterations in lamina k , but with damaged properties calculated according to the current values of their damage variables $D_i^{(m)}$. Given a trial value of λ_k , the analytical solution provides $g_k, D_i^{(k)}$ for lamina k assuming all other laminas do not damage while performing lamina iterations in lamina k . Since the solution for lamina k depends on the stiffness of the remaining laminas, a converged iteration for lamina k does not guarantee convergence for the same lamina once the damage in the remaining laminas is updated. In other words, within a given strain step, the stiffness and damage of all the laminas are interrelated and they must all converge. This can be accomplished by laminate-iterations; that is, looping over all laminas repeatedly until all laminas converge to $g = 0$ for all k .

Chapter 4

Methodology

In this section, it is described a methodology to predict damage initiation and evolution as function of service conditions. The proposed method is useful to obtain material properties of any type of materials. In concrete, we focus on getting the material properties for damage analysis of Glass and Carbon fiber composites through of the progressive damage analysis (PDA) model already implemented in Abaqus and explained in Chapter 2.

Although elastic moduli data and standard experimental methods exist, such data and methods do not exist for damage-related properties. However, such experimental data display macroscopic effects of damage (e.g., crack density and stiffness reduction) for a number of material systems.

In order to predict the material response (damage) for Glass and Carbon fiber composite laminates using Abaqus, we must obtain first the missing material properties used by PDA which predict and match with experimental data. Therefore, our methodology will based on minimizing the error between predictions and available experimental data.

4.1 Abaqus Script

Abaqus will be used to predict the laminate stiffness reduction as function of applied strain using PDA. Then, such results are compared with the type of data available from experiments where will be normally as function of strain. The PDA model is used for all the composite laminas in the laminate, and thus replacing the standard linear material model that it is usually employed in Finite Element Analysis (FEA). The geometry used will resemble the gauge section of the uniaxial tensile specimens used in the experiments. Since all the laminates included in the experimental studies are balanced symmetric, only a quarter of the specimen is modeled by using symmetry boundary conditions. A single element is generally used. Only a few cases are meshed with different number and type of elements to study mesh-sensitivity later. For each simulation, a uniform strain is applied via imposed displacements at one end of the specimen.

Lamina orthotropic elastic properties were obtained from the literature. An example is shown in Appendix A.

4.2 Determination of material properties

Once we have simulated the damage initiation and evolution through of the Abaqus script, the question arises how to obtain the material properties. The following approach is then introduced. Given the available experimental data, a least square problem is solved minimizing the error at each point between the experimental data and PDA values, in order to fit as much as possible with the results. This procedure is carried out for each material to be characterized and once this optimization problem is solved, it is used to predict and compare with different laminates of the same material system. In this way, it is compared the quality of PDA model to predict damage initiation and evolution.

In this work, we will focus in describing the matrix cracking produced during its first life step, namely the first type of damage that appears in composite laminates. With this objective, we center in the main values which produces de initiation of damage, namely *in-situ strength* of unidirectional lamina for transverse strength (F_{2t}^{is}) and shear strength (F_6^{is}), and the energy dissipation property G_{mt}^i for matrix tension as it was explained in Chapter 2. With these three parameters we should predict efficiently the damage initiation and evolution given an applied strain. Note that as it was explained in Chapter 2, this energy dissipation property G_{mt}^i does not correspond with the critical Energy Release Rate (ERR) described in Chapter 3. The critical ERR is a material property that take into account a new crack surface area while the energy dissipation property G_{mt}^i is an unidimensional parameter which simulates a loss of lamina stiffness through of reduction stiffness coefficients d_m^t of each lamina.

The majority of experimental data used in the literature were shown as Normalized Young modulus vs. applied strain. For this reason, all the values were compared using the same pattern. Such results were obtained from the Abaqus script through the linear elastic equation $\sigma = E \epsilon$, since the whole laminate is homogenized with the stiffness coefficients d_m^t . While the applied strain is imposed, the stress at each displacement is calculated dividing the sum of nodes's reaction forces by the total area as shown the Figure 4.1. Using the specimen geometry sections, the area is calculated as the laminate width by the total thickness (included the symmetric part). As it can be seen in Figure 4.1, the laminate's stiffness decreases after some strain reducing the chart slope. Note that the slope change corresponds to the damage initiation once the Hashin's criteria has been satisfied.

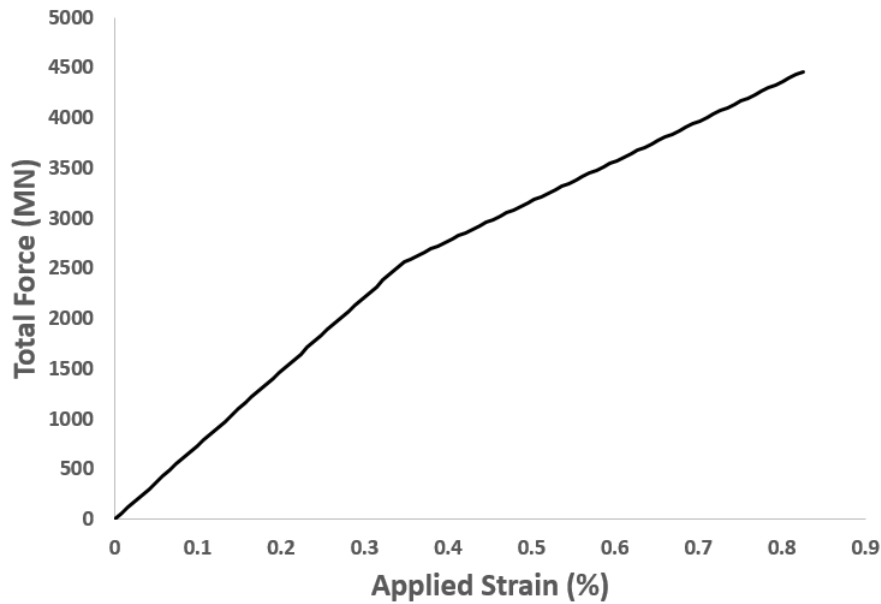


Figure 4.1: Force vs. displacements for a laminate with loss of stiffness.

4.3 Optimization problem

The methodology to solve an optimization problem, which is used to calculate the material properties needed, are presented below. Due to the idiosyncrasies of each commercial software, there are two possible methods.. Each one has its advantages and disadvantages, and they will be noted.

4.3.1 MATLAB® script controlling Abaqus

The first possibility is to carry out the optimization problem by using a MATLAB® Script. The main advantage is that MATLAB is highly efficient to handle mathematical data. For our purpose, MATLAB has to call Abaqus in a command windows and run a predefined script that completely defines the FEA model to be solved as was explained in Section 4.1. This script contains the necessary commands to create the laminate, specimen geometry, boundary conditions (bc), and load it with an incremental displacement to simulate applied strain. The script is developed in Python version 2.7 compatible with Abaqus version 6.14-2. The main steps are:

- Import all the libraries customized for Abaqus and set the work directory in which all files or jobs will be kept.
- Update the variables to be used in each iteration as requested by the MATLAB code. MATLAB overwrites the Python script in each iteration.
- Then, set the known material properties, section lay-out, assign, assembly, step, bc, meshing, and create an Abaqus Job. In Abaqus module *step*, the applied displacement is split

in a fixed number (100, 500, or 1500) to get predictions that correspond closely to experimental data points.

- Finally, the script submits the job to get results, which are saved to a report file, to be read by and post processed by MATLAB.

A MATLAB script computes the error (1) using Abaqus results and experimental data. A brief description of such script follows:

- An Excel file with experimental results is created, to be read by MATLAB.
- A function collect the experimental data in an array.
- A "handle" function runs the Abaqus script and forces PDA results to match laminate stiffness degradation data. The Abaqus script is run from MATLAB. The error between predicted and experimental data is defined as follows:

$$Error = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\left. \frac{E}{\hat{E}} \right|_{\epsilon=\epsilon_i}^{Abaqus} - \left. \frac{E}{\hat{E}} \right|_{\epsilon=\epsilon_i}^{Experimental} \right)^2} \quad (4.1)$$

- Finally, the script uses these functions and sets the initial guess points ($x_o = [G_c, F_{2t}]$) the function arguments, the initial constrains, and the optimization settings in MATLAB.

Convergence speed with number of iterations is shown in Figure 4.2. It can be observed that the first points are equivalent to the Response Surface Optimization (fast estimation of the exact solution) and the rest to a Direct Optimization as the properties converge to the exact solution.

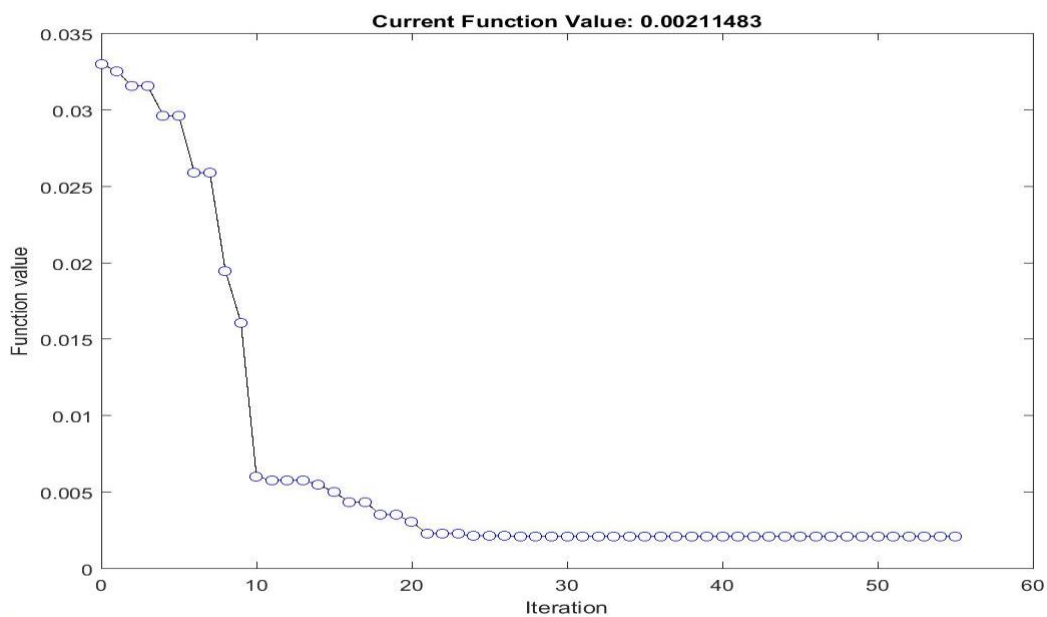


Figure 4.2: Convergence of the error vs. the number of iterations with fminsearch function..

4.3.2 Abaqus® script

The second proposed methodology is using only an Abaqus script. This method requires higher knowledge of Python from the programmer but it is simpler for the user. Furthermore, there is no need for MATLAB and execution is faster.

Installing Python and its corresponding math libraries is problematic. Math libraries (Scipy and Numpy) are needed to perform elemental and advanced math operations required for optimization. Installation of these libraries to extend Abaqus functionality is described next:

- Determine the Python (`>>> import sys`) and Numpy version (`>>> import numpy; numpy.version.version`) for the installed Abaqus release.
- Install the correct Python version on the computer. Onwards, any necessary library to be used by Abaqus except Numpy library must be first installed in Python folder. Note that Numpy library is already installed in Abaqus by default (Numpy version cannot be changed).
- Install the Scipy library version that works with the Numpy version already installed by Abaqus in the correct OS (32 or 64 bit) and by default is installed in the folder of Python. For instance, Abaqus 6.14 (64 bit) works with Python 2.7, which works with the Scipy library compatible with Numpy 1.6.2 (64 bit). This Scipy library contains the optimization and advanced mathematical functions.
- Once it has been done this, the scipy library installed in Python folder, it is copied and moved to the Abaqus library folder (C:\SIMULIA\Abaqus\6.14-2\tools\SMAPy\python2.7\Lib\site-packages\scipy). After that, it is possible to import any function from scipy library. This can be done for any others kind of libraries.

The rest of the problem is limited to write three Abaqus scripts and run them directly from CAE. Each script plays an important part and must be in the same folder as the Abaqus work directory. Then, each script can be called through import command. An example is shown in Appendix A. The function of each script are described as follows:

- In the first script, all Abaqus customized libraries are imported. A new class model, which constructs functions to set up, run, and get results from Abaqus, is defined. The first two functions set up the laminate specimen as well as properties, composite lay-out, bc, meshing, and creates an Abaqus job. The last function gets the predicted response XY data. Note that these results are kept in a temporary file that must be erased at the end of each iteration to avoid storing multiple results with same variable name.
- The objective error function through the equation (4.1) to match laminate stiffness degradation data is defined in the second script. Note that the experimental data are written inside the code to avoid wasting time and the class model is imported.

- Finally in the third script, the error function is imported and the optimization problem is solved using the *fmin* function from the Scipy library (equivalent to *fminsearch* in MATLAB). Then, options and tolerances are adjusted, and the results (function value, number of iterations and function evaluations) are printed on the Abaqus window. A mapping of the exact solution through of *fmin* function with *scipy* is illustrated in Figure 4.3.

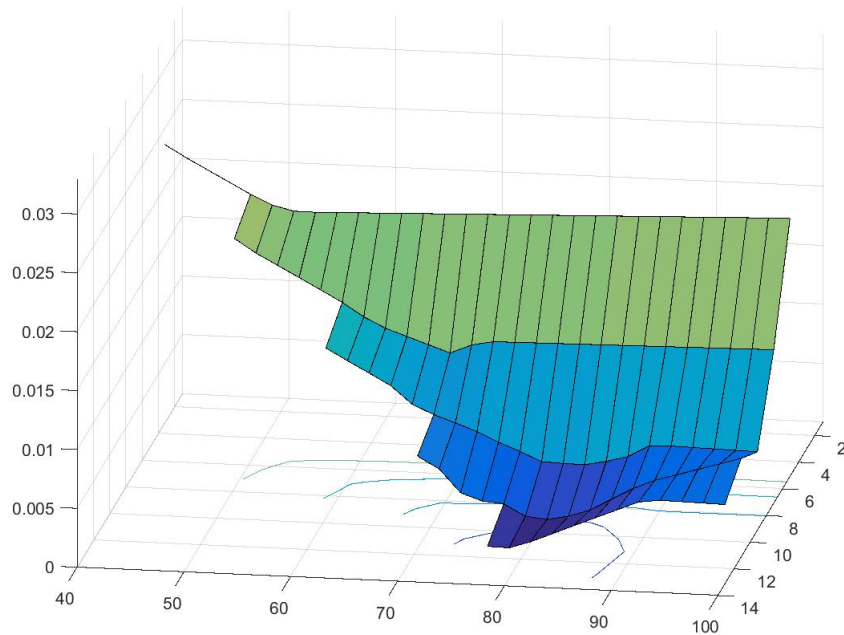


Figure 4.3: Convergence of the error vs. the number of iterations with *fmin* function.

4.4 Generate Modulus Reduction Data

Until now, the purpose of this study is to develop a method to obtain the missing material properties by adjusting their values so that the predicted material response matches experimental data. Once the material properties are obtained, the simulation predictions can be carry out to compare with a broad set of glass and carbon composite laminates with different laminate stacking sequences. Initially, the proposed method relies on availability of measured stiffness reduction vs. applied load or strain. As it was pointed in [31], such data is easy to obtain for glass/polymer composites, where the stiffness of the matrix has a noticeably effect on the stiffness of the composite, but carbon fibers are so stiff that the degradation of the matrix can go unnoticed in stiffness reduction measurements. A more direct measure of damage, i.e, crack density (cracks/mm), is often reported in the literature [3, 4, 10, 32–35, 35, 36], but PDA does not calculate crack density and thus cannot be compared directly to crack-density data. To solve this problem, a novel data processing method is proposed to derive stiffness reduction in terms of available crack-density data using an intermediate damage mechanics model (DDM) [22, 37, 38]

that use crack density data to predict stiffness reduction explained in Chapter 3. Then, the derived stiffness reduction data is used to obtain the material properties needed for using PDA in Abaqus.

In this section, experimental data of crack density vs. applied strain or applied stress is summarized and used to calculate the damage material parameters for the discrete damage mechanics DDM model as well as to generate the modulus reduction data that it is needed to adjust the PDA parameters in Chapter 2. Measuring changes of longitudinal modulus vs. applied strain is almost impossible for carbon-based composites because the stiffness of the fibers dominates the stiffness of the laminate regardless of what happens to the matrix. Even if the crack density is high, the load drop is supported by the fibers which barely change countenance. In addition, experimental errors are of the same order of magnitude of the stiffness reduction, thus compromising the reliability of experimental data. Instead, different alternatives to characterize damage progression in Carbon/polymer composites have been reported in the literature [3, 4, 32, 33]. A comparison between glass and carbon laminates with same stacking sequence is shown in Figure 4.4 to illustrate magnitude different of stiffness reduction. It can be seen that the loss of stiffness for glass/epoxy laminates is enough higher than for carbon/epoxy laminates in general. The modulus drops 8.75% (from 58.03 to 52.95 GPa) for $[0_2/90_4]_s$ IM7/MTM45 and 26.9% (from 23.54 to 17.21 GPa) for $[0_2/90_4]_s$ Fiberite/HyE-9082A. Although increasing and decreasing the 90° and 0° laminas respectively we get a larger loss of stiffness and thus more easily to be measured [38], not useful experiments comparing the Young Modulus vs. strain was found in the literature.

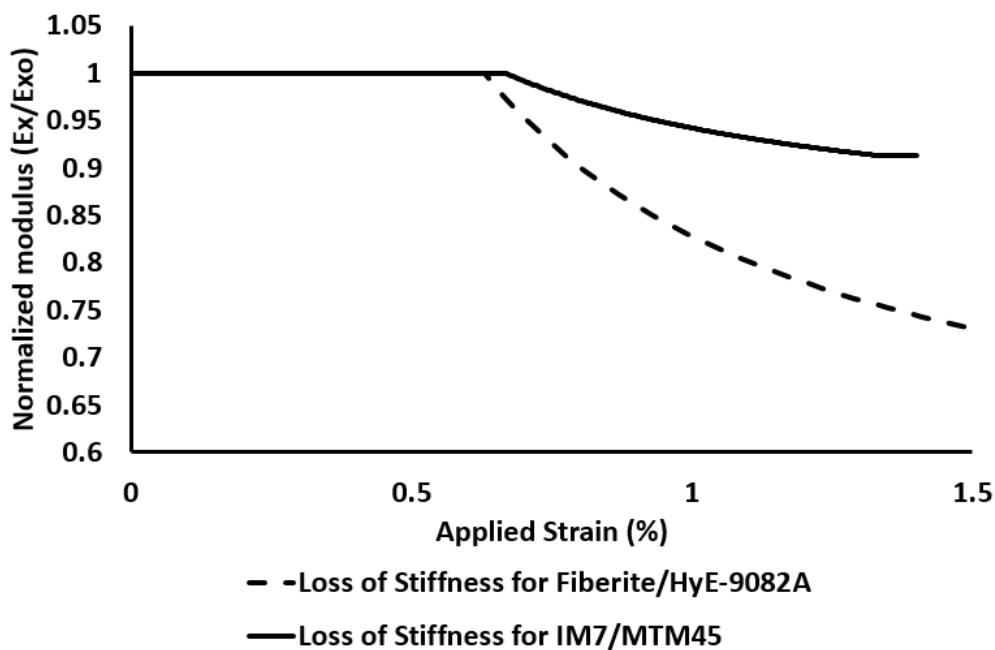


Figure 4.4: A comparison of Normalized Young Modulus between glass and carbon laminate with same LSS: $[0_2/90_4]_s$.

Crack density λ vs applied strain ϵ_x or applied stress $\sigma_x = N_x/t$, has been measured experimentally. None of the laminas in the laminates measured experimentally were subjected to fiber or matrix mode compression (2.2, 2.3). However, crack density cannot be compared and matched with PDA predictions because PDA does not calculate crack density. The solution proposed herein is to calculate stiffness reduction data using another constitutive model that calculates crack density and thus can be compared to crack density data, namely DDM model described in Chapter 3. Once the stiffness reduction data has been generated, it is then used to adjust the material properties for PDA through an optimization process.

A discrete damage mechanics model (DDM) [22] was chosen to generate stiffness reduction data. DDM is implemented as a user general section (UGENS) in Abaqus [23]. The damage parameters of the discrete damage mechanics (DDM) constitutive equation can be adjusted by comparing DDM predictions of crack density with experimental crack-density data [4]. The material property determination for DDM is performed by executing a Python optimization script Appendix [] that adjusts the properties to minimize the error between predictions and experimental crack density data. The script runs inside Abaqus/CAE to calculate crack density and stiffness reduction, via DDM, as required by the optimization algorithm. The DDM error is calculated as follows

$$Error = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\lambda|_{\epsilon=\epsilon_{-i}}^{Abaqus} - \lambda|_{\epsilon=\epsilon_{-i}}^{Experimental} \right)^2} \quad (4.2)$$

where λ^{Abaqus} , $\lambda^{Experimental}$, are the predicted and measured crack density, and N is the number of data points.

DDM requires just two properties, the true energy release rates (3D ERR) G_{Ic} and G_{IIc} . Note that the fracture energy G_c in PDA is not true 3D ERR and thus it is not numerically equal to G_{Ic} or G_{IIc} or any combination thereof. The reason for this discrepancy is due to the fact that PDA transforms the 3D problem into a 1D problem by equations (2.11, 2.14). When laminates with 0° or 90° laminas are subjected to uniaxial extension, the laminas are not subjected to any shear, so damage initiation and evolution are both controlled by G_{Ic} (crack opening mode I). On the other hand, when laminates with $\pm\theta$ laminas are subjected to uniaxial extension, both traction and shear may appear, so damage initiation and evolution are controlled by both G_{Ic} and G_{IIc} . DDM not only calculates the crack density but also the stiffness reduction of the laminate, thus providing the data needed for adjusting the properties needed for PDA. Once the stiffness reduction data is generated, the PDA properties can be adjusted by minimizing the PDA error as equation (4.1), namely

$$Error = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\frac{E}{\hat{E}} \Big|_{\epsilon=\epsilon_{-i}}^{Abaqus} - \frac{E}{\hat{E}} \Big|_{\epsilon=\epsilon_{-i}}^{Experimental} \right)^2} \quad (4.3)$$

All the laminates considered for the study are symmetric and balanced. Therefore a quarter of the specimen was used for the analysis using symmetry b.c. and applying a uniform strain via imposed displacements on one end of the specimen. The dimensions of the specimens are 12 mm wide with a free length of 150 mm. Furthermore, the DDM predictions are insensitive to the type of element used, namely linear S4R or quadratic S8R. The ply material properties are listed in Table 1.

Tables 4.1: Unidirectional ply properties for laminates IM6/Avimid K, T300/Fiberite 934, AS4/Hercules 3501-6 and IM7/MTM45-1 [1-4].

Property	IM6/Avimid K	T300/Fiberite 934	AS4/Hercules 3501-6	IM7/MTM45-1
E_1 [GPa]	134.0	128.0	130.0	157.9
E_2 [GPa]	9.8	7.2	9.7	7.7
G_{12} [GPa]	5.5	4.0	5.0	3.6
G_{23} [GPa]	3.6	2.4	3.6	2.7
ν_{12}	0.300	0.300	0.30	0.360
ν_{23}	0.361	0.501	0.347	0.400
α_1 [$\mu\epsilon/K$]	-0.09	-0.09	-0.09	-5.5
α_2 [$\mu\epsilon/K$]	28.8	28.8	28.8	28.5
Ply thickness [mm]	0.144	0.144	0.144	0.14
F_{1t} [MPa]	2326	1500	1950	2465
F_{2t} [MPa]	37	27	48	52
F_{1c} [MPa]	1000	900	1480	1252
F_{2c} [MPa]	200	200	200	193
F_6 [MPa]	63	100	79	48

4.4.1 IM6/Avimid ®K Polymer

Crack density λ vs applied stress σ_x measured experimentally [4] are used to indirectly adjust the PDA model. Laminate 2 in Table 4.2 was used to adjust G_{Ic} because it shows the strongest mode I fracture behavior of the group [5]. The calculated value G_{Ic} is reported in Table 4.3. Note that not laminates subjected to pure shear are reported in [4], so G_{IIc} cannot be calculated. Comparison between crack density predicted by DDM and experimental data is shown in Figure 4.5. Simultaneously, the modulus reduction generated by DDM for $[0/90_3]_s$ laminate IM6/Avimid K Polymer is reported in Figure 4.6 denoted as "*generated modulus reduction data*" to be compared with PDA later.

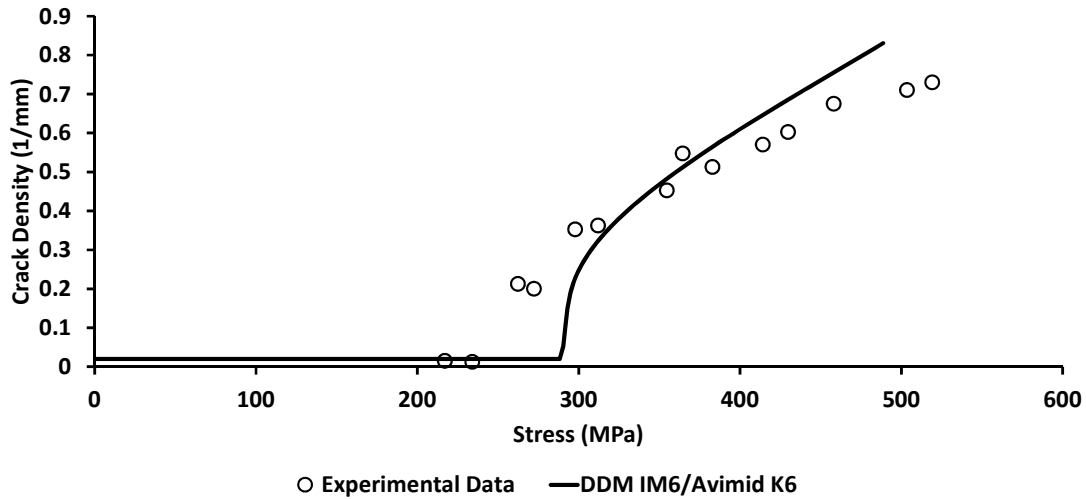


Figure 4.5: DDM model prediction and crack density data in $[0/90_3]_s$ laminate IM6/Avimid®K Polymer.

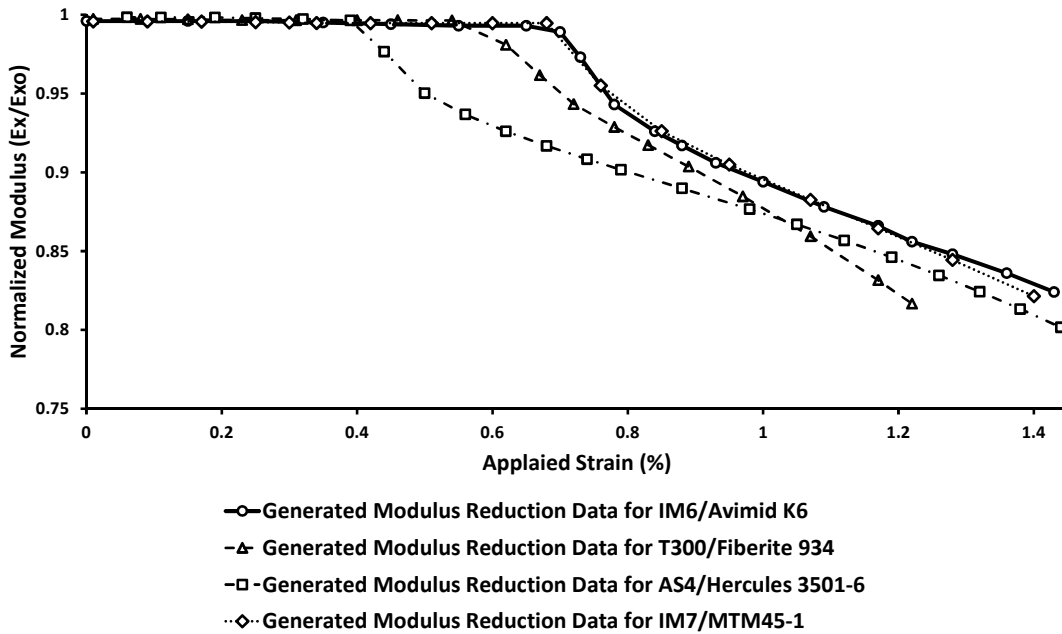


Figure 4.6: Modulus reduction generated by DDM for several material systems: $[0/90_3]_s$ laminate IM6/Avimid®K Polymer, $[0/90_4]_s$ laminate T300/Fiberite 934, $[0/90_2]_s$ laminate AS4/Hercules 3501-6 and $[0/90_4]_s$ laminate IM7/MTM45-1.

4.4.2 T300/Fiberite 934

Crack density λ vs applied stress σ_x measured experimentally [4] are used to indirectly adjust the PDA model. Laminate 6 in Table 4.2 was used to adjust G_{Ic} because it shows the strongest mode I fracture behavior of the group [5]. The calculated value G_{Ic} is reported in Table 4.3. Note that not laminates subjected to pure shear are reported in [4], so G_{IIc} cannot be calculated. A strain of approximately 1.16 % was found to break some fibers in 0° degrees laminas so, data points beyond that strain were not used to generate stiffness reduction data. Comparison between crack density predicted by DDM and experimental data is shown in Figure 4.7. Simultaneously,

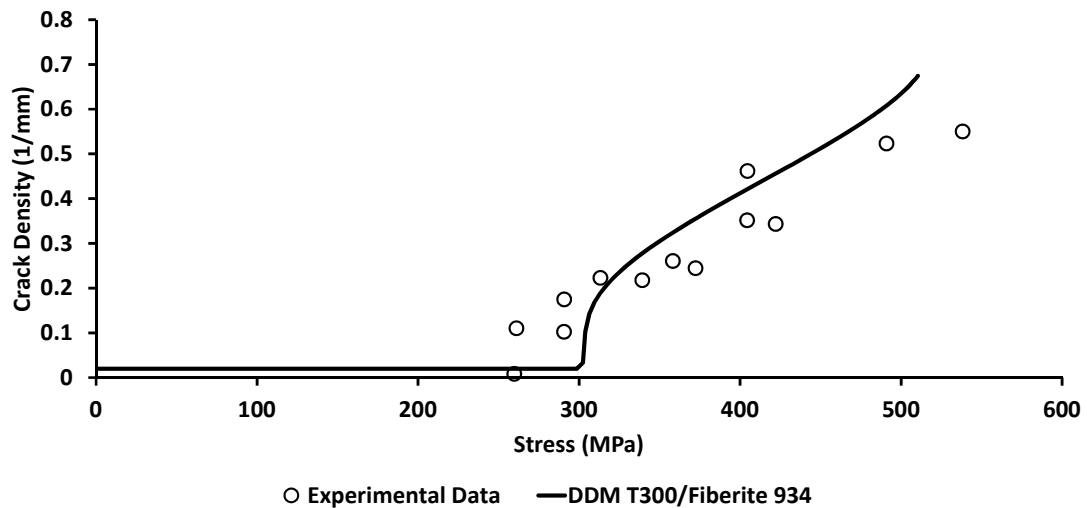
Tables 4.2: Laminates considered in this study.

Laminate	Stacking Sequence	Material
1	$[0/90_2]_s$	IM6/Avimid K [4]
2	$[0/90_3]_s$	
3	$[0_2/90_2]_s$	
4	$[0_2/90_4]_s$	
5	$[0/90_2]_s$	T300/Fiberite 934 [4]
6	$[0/90_4]_s$	
7	$[0_2/90]_s$	
8	$[0_2/90_2]_s$	
9	$[0_2/90_4]_s$	
10	$[0/90_2]_s$	AS4/Hercules 3501-6 [4]
11	$[0_2/90_2]_s$	
12	$[0/90_4]_s$	IM7/MTM45-1 [3]
13	$[\pm 25/90_5]_s$	
14	$[0/\pm 55_4/0_{1/2}]_s$	
15	$[0/\pm 70_4/0_{1/2}]_s$	

Tables 4.3: Critical energy release rates for Discrete Damage Mechanics (DDM) model [5].

DDM Parameters	IM6/Avimid K	T300/Fiberite 934	AS4/Hercules 3501-6	IM7/MTM45-1
G_{Ic} [J/m^2]	258.0	208.0	60.0	255.1
G_{IIc} [J/m^2]	-	-	-	598.1

the modulus reduction generated by DDM for $[0/90_4]_s$ laminate T300/Fiberite 934 is reported in Figure 4.6.


 Figure 4.7: DDM model prediction and crack density data in $[0_2/90_4]_s$ laminate T300/Fiberite 934.

4.4.3 AS4/Hercules 3501-6

Crack density λ vs applied stress σ_x measured experimentally [4] are used to indirectly adjust the PDA model. Laminate 10 in Table 4.2 was used to adjust G_{Ic} because it shows the strongest mode I fracture behavior of the group [5]. The calculated value G_{Ic} is reported in Table 4.3. Note that not laminates subjected to pure shear are reported in [4], so G_{IIc} cannot be calculated. Comparison between crack density predicted by DDM and experimental data is shown in Figure 4.8. Simultaneously, the modulus reduction generated by DDM for $[0/90_4]_s$ laminate AS4/Hercules 3501-6 is reported in Figure 4.6.

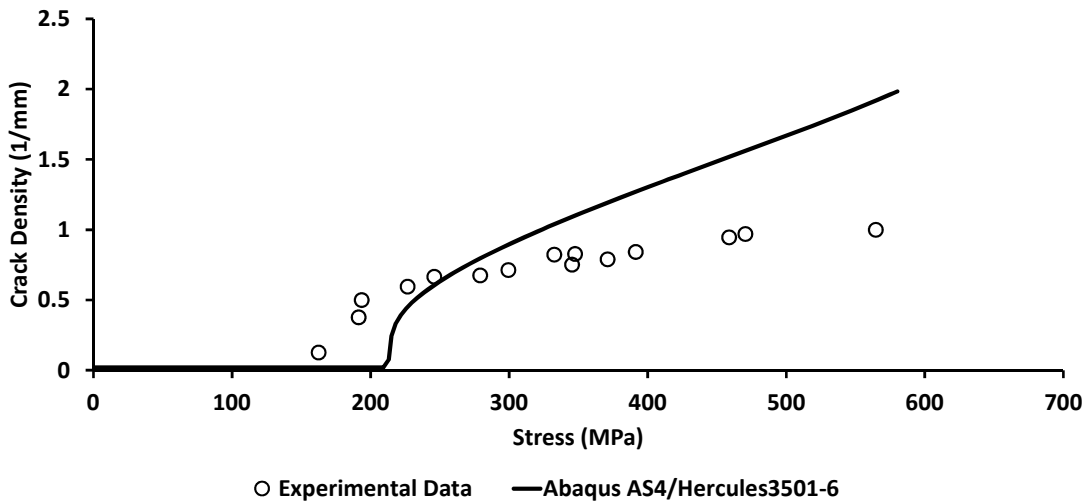


Figure 4.8: DDM model prediction and crack density data in $[0/90_2]_s$ laminate AS4/Hercules 3501-6.

4.4.4 IM7/MTM45-1

Crack density λ vs applied strain ϵ_x measured experimentally [3] are used to indirectly adjust the PDA model. Laminate 12 in Table 4.2 was used to adjust G_{Ic} because it shows the strongest mode I fracture behavior of the group. In the same way, laminate 14 was chosen because it shows almost pure shear, so the strongest mode II fracture behavior. The DDM properties that yield the best match to crack density data are reported in Table 4.3. Comparison between crack density predicted by DDM and experimental data [3] is shown in Figure 4.9 and 4.10 for $[0/90_4]_s$ and $[0/\pm 55_4/0_{1/2}]_s$ laminate IM7/MTM45-1, respectively. The dimensions of these specimens are 19 mm wide with a free length of 270 mm. The ply material properties are listed in Table 4.1. Simultaneously, the modulus reduction generated by DDM for $[0/90_4]_s$ laminate IM7/MTM45-1 is illustrated in Figure 4.6.

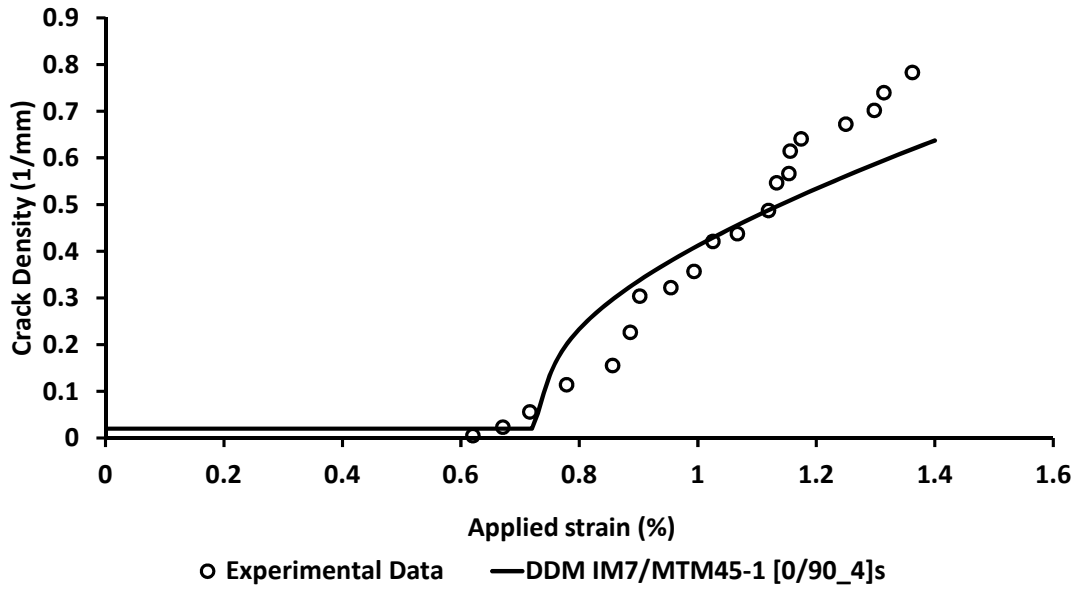


Figure 4.9: DDM model prediction and crack density data in $[0/90_4]_s$ laminate IM7/MTM45-1.

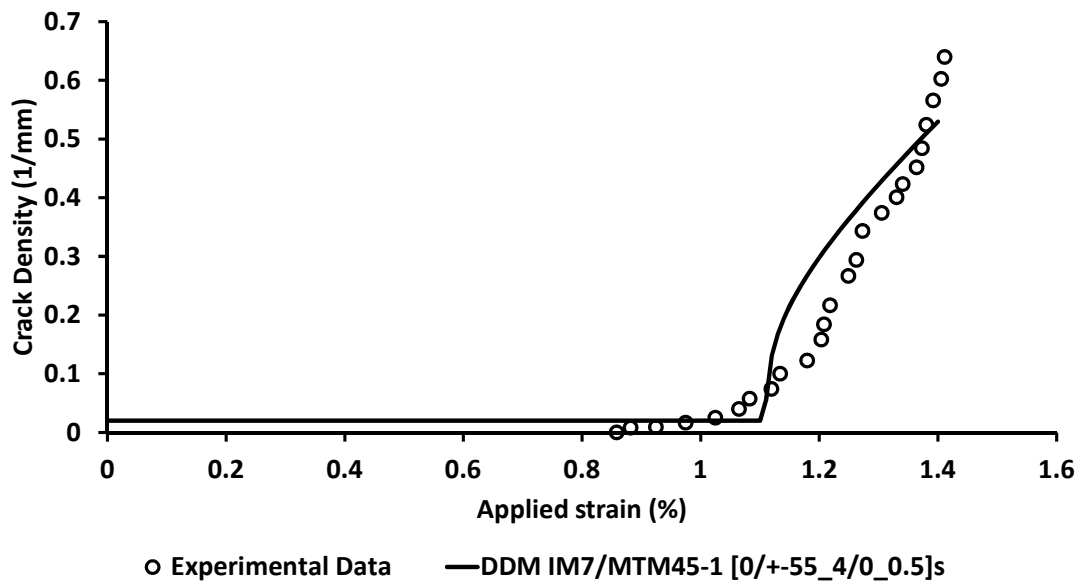


Figure 4.10: DDM model prediction and crack density data in $[0/\pm 55_4/0_{1/2}]_s$ laminate IM7/MTM45-1.

Chapter 5

Results and Adjusted Parameters

In this section, the material missing properties are adjusted and the results are shown. Such results are split for each material system, glass and carbon composite laminates. In concrete, a brief description for E-glass epoxy laminates is presented to illustrate the methodology used described previously as well as the adjusted parameters for Abaqus 6.14. Then, the carbon epoxy laminates are shown using the generated modulus reduction data through of DDM model with the corresponding material properties for each material system. Later, a comparison between the adjusted and calculated in-situ strength values are compared. Finally, a convergence and mesh sensitivity is studied.

5.1 E-glass epoxy laminate

The material system used is Fiberite/HyE-9082A [32,33]. A $[0_2/90_4]_s$ laminate is analyzed first to adjust the energy dissipation property G_{mt}^c and the in-situ transverse tensile strength F_{2t}^{is} by minimizing the error between predicted and experimental values of laminate stiffness. Any laminate with the laminas $\pm\theta$ close to $\pm 45^\circ$ is useful to optimize the in-situ shear strength F_6^{is} while G_{mt}^c and F_{2t}^{is} are kept constant at the values obtained previously. For all cases, the final results are similar Table 5.1. The number of points at which the laminate stiffness is calculated has a little impact on the accuracy. Note that F_6^{is} and F_6^{is} are the in-situ strength values since the force that we need to apply in order to produce the first crack increases once the lamina is embedded within a laminate.

Normalized modulus versus the applied strain for both laminates are shown in Figure 5.1

Tables 5.1: PDA optimal parameters calibrated with different elements.

Property	Units	S4R	S8R
F_{2t}	[MPa]	80.0625	79.2788
F_6	[MPa]	50.0024	50.3185
G_{mt}^c	[kJ/m]	26.1875	12.2861

and 5.2. In laminate $[0_2/90_4]_s$ Abaqus results fit the experimental data nicely. For the laminate $[0/\pm 40_4/0_{1/2}]_s$, the average error is larger. The discrepancy can be explained as follows. PDA does not include dissipation energy due to in-plane shear separately from transverse tension, but rather both are lumped into one term G_{mt}^c which is bad and erroneous formulation.

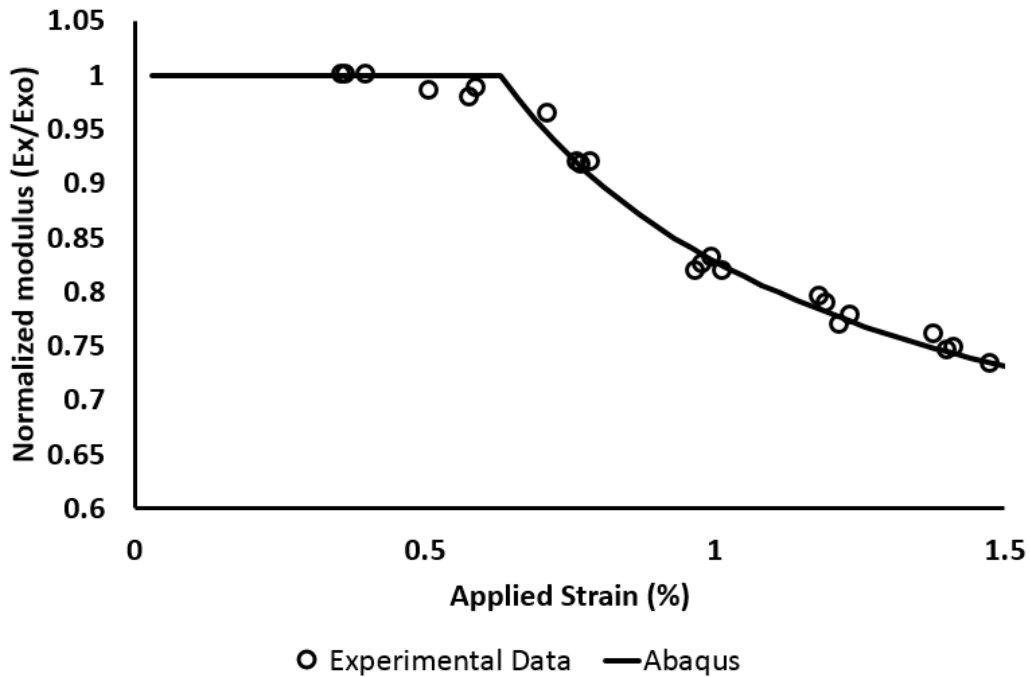


Figure 5.1: Normalized modulus vs. applied strain for laminate $[0/90_4]_s$ Fiberite/HyE-9082A.

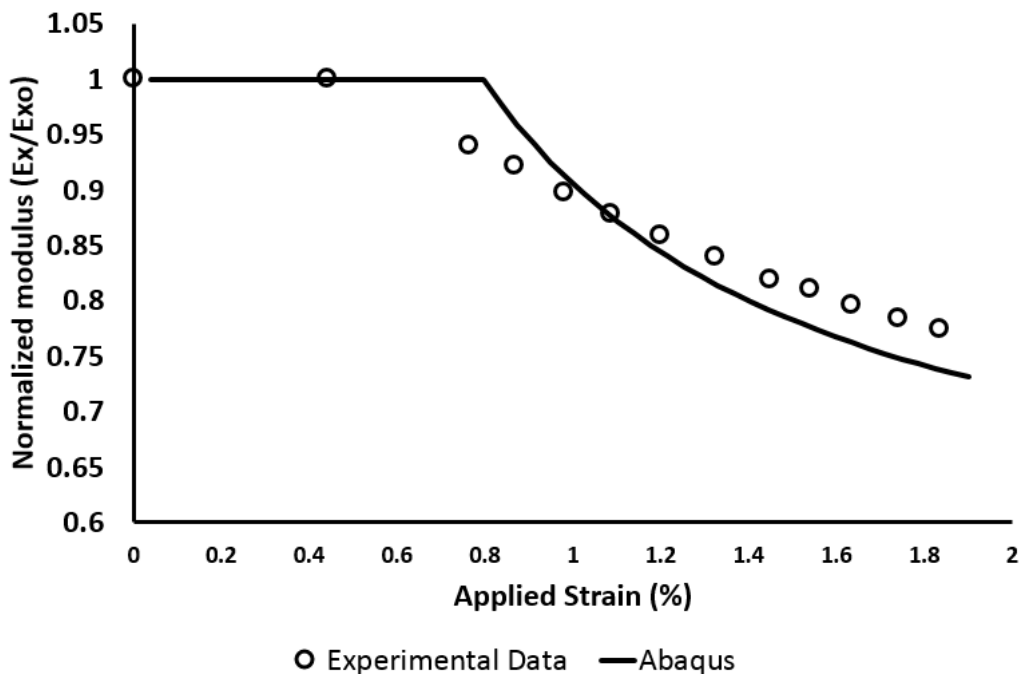


Figure 5.2: Normalized modulus vs. applied strain for laminate $[0/\pm 40_4/0_{1/2}]_s$ Fiberite/HyE-9082A.

5.2 Carbon epoxy laminate

The material systems used are IM6/Avimid K, T300/Fiberite 934, AS4/Hercules 3501-6 and IM7/MTM45-1 [1–4]. The stiffness reduction data generated in Section 4.4 is used to adjust the damage parameters in PDA. That is, the in-situ transverse tensile strength F_{2t}^{is} , the in-situ shear strength F_6^{is} , and the PDA fracture energy G_{mt}^c are determined so that the PDA prediction are as close as possible to the data. For IM7/MTM45-1 [3], the $[0/90_4]_s$ laminate is analyzed first to adjust F_{2t}^{is} and G_{mt}^c by minimizing the error between predicted and semi-empirical experimental values of laminate stiffness reduction. Next, any laminate with laminae $\pm\theta$ close to $\pm 45^\circ$ is useful to adjust F_6^{is} while F_{2t}^{is} and G_{mt}^c are kept constants. A $[0/\pm 55_4/0_{1/2}]_s$ laminate was used to adjust F_6^{is} . Results are shown in Table 5.2. The number of generated semi-empirical data points has a little impact on the values obtained for the properties as long as the stiffness degradation is well represented.

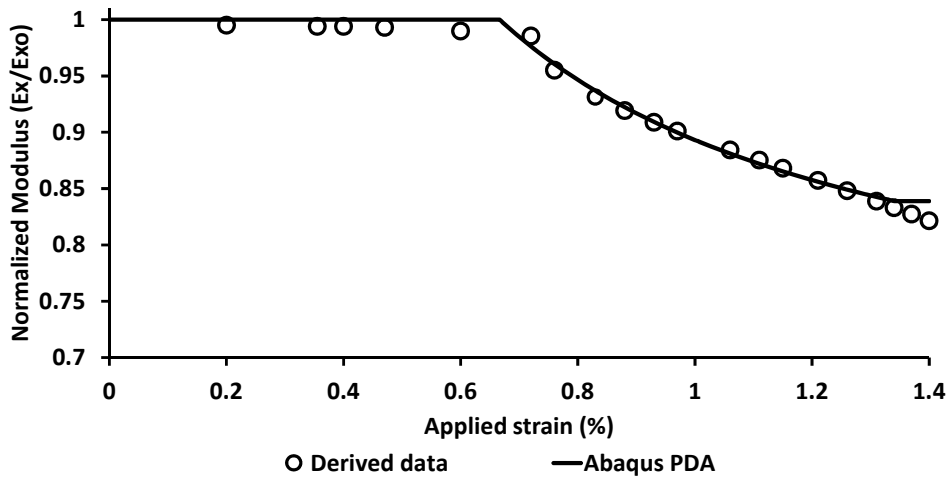


Figure 5.3: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_4]_s$ IM7/MTM45-1.

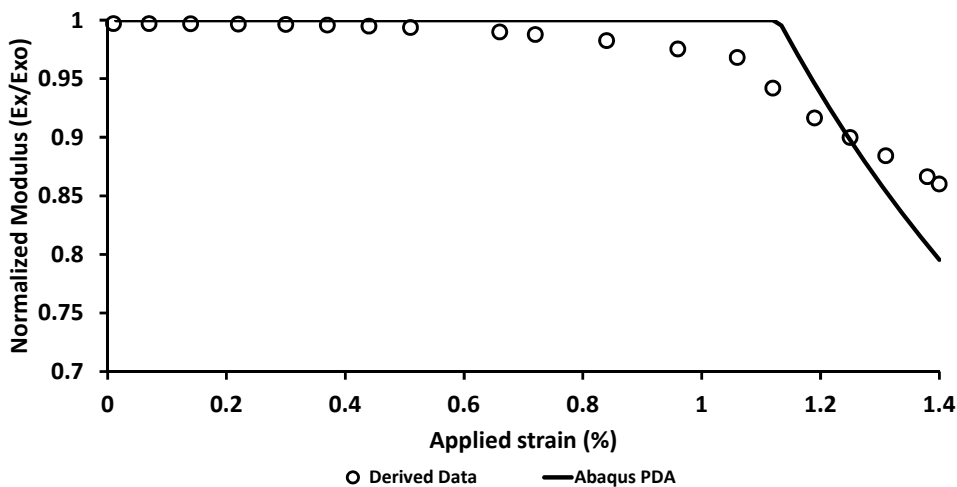


Figure 5.4: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode II for $[0/\pm 55_4/0_{1/2}]_s$ IM7/MTM45-1.

Tables 5.2: In-situ strengths and PDA fracture energy G_{mt}^c .

Property	Units	IM6/Avimid K		T300/Fiberite 934		AS4/Hercules 3501		IM7/MTM45	
Carbon/Epoxy		S8R	S4R	S8R	S4R	S8R	S4R	S8R	S4R
F_{2t}^{is}	[MPa]	64.5926	64.5896	41.2506	41.3186	37.2957	37.2320	50.9081	50.9103
F_6^{is}	[MPa]	-	-	-	-	-	-	128.4921	128.8867
G_{mt}^c	[KJ/m]	10.3045	21.1528	3.9838	8.1693	4.8687	9.7574	12.2171	21.1091

Modulus versus the applied strain are shown in Fig. 5.3 and 5.4 for two IM7/MTM45-1 laminates. For laminate $[0/90_4]_s$ in Fig. 5.3, PDA results fit the experimental data nicely. For $[0/\pm 55_4/0_{1/2}]_s$ in Fig. 5.4, the error is larger. The discrepancy can be explained again due to absence of a dissipation energy for shear mode separately from transverse tension since it uses the same term G_{mt}^c . Thus PDA struggles when both modes, traction and shear are combined in $[0/\pm 55_4/0_{1/2}]_s$. Note also that the in-situ shear strength F_6^{is} should match the onset of laminate modulus reduction data. Instead, F_6^{is} is adjusted to start cracking a little late. This is because the optimization function seeks to reduce the error (4.2) between the data points and the PDA prediction. Thus, increasing slightly the F_6^{is} value, we reduce the error produced by the rest of experimental points which belong to the AC side in Fig. 2.1. This would not happen if the dissipation energy due to shear were computed separately from G_{mt}^c .

In the same way, modulus versus applied strain are shown in Fig 5.5-5.7 for IM6/Avimid. Laminate 2 (Table 4.2) was chosen to fit PDA results with the experimental data as is illustrated in Fig. 5.5. For T300/Fiberite 934, laminate 6 was chosen to fit with experimental data as is shown in Fig. 5.6. For AS4/Hercules 3501, laminate 10 was chosen to fit with experimental data as is shown in Fig. 5.7.

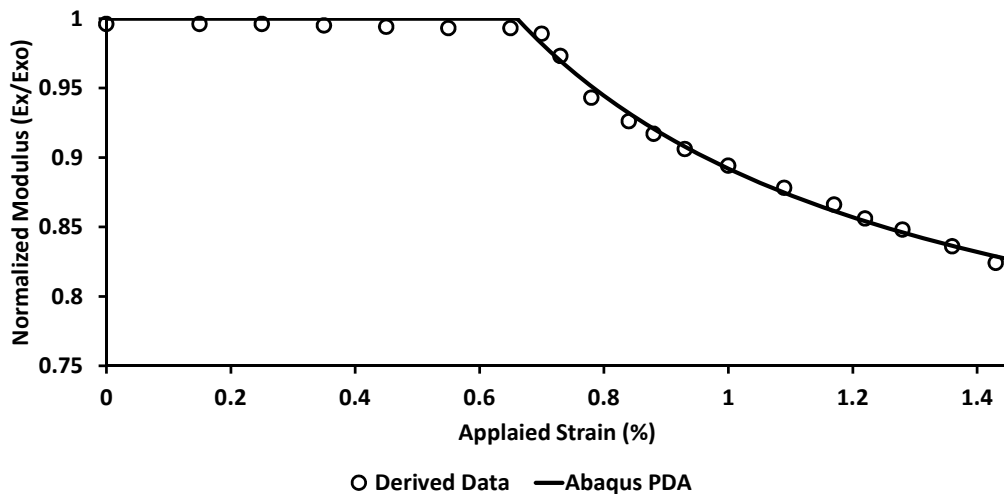


Figure 5.5: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_3]_s$ IM6/Avimid®K Polymer.

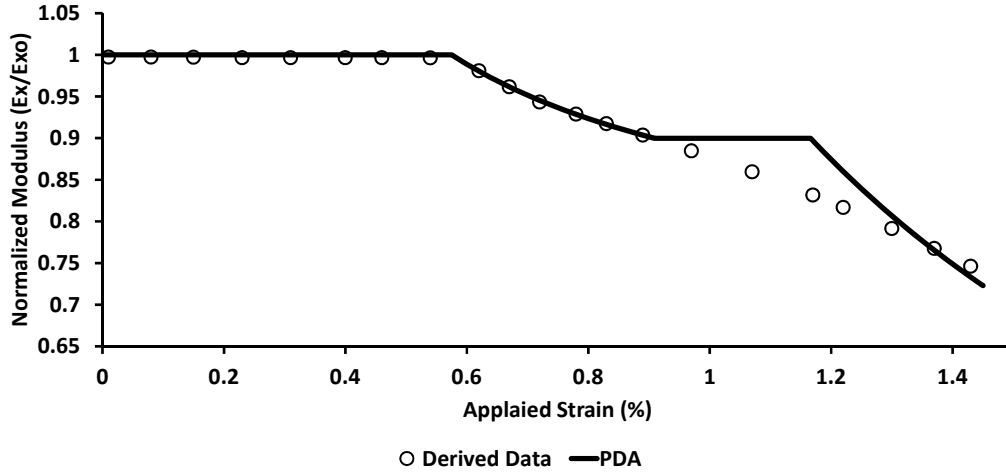


Figure 5.6: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_4]_s$ T300/Fiberite 934.

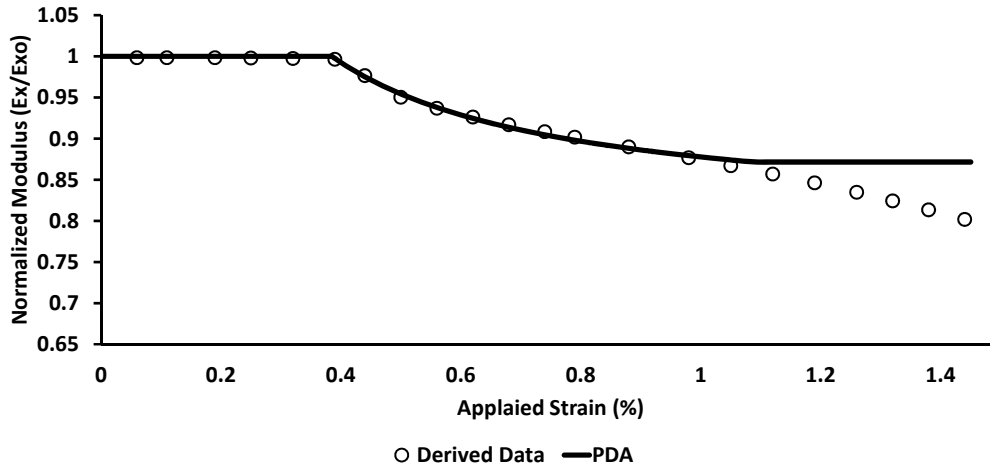


Figure 5.7: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_2]_s$ AS4/Hercules 3501.

5.3 In-situ strength

The in-situ strength values can be obtained directly from laminate data as it is shown in section 5.2. Alternatively, they could be calculated with the methodology presented in [6, section 7.2.1]. Assuming a transition thickness $t_t = 0.8$ mm, for each of the four carbon material systems, the following calculations are performed and the results are shown in Table 5.3. The effective thickness is

$$t_e = \min(t_t, t_k) \quad (5.1)$$

where t_k is the thickness of the cracking lamina. Then, the in-situ values are calculated as follows:

$$F_{2t}^{is} = 1.12 F_{2t} \sqrt{\frac{2 t_t}{t_e}} \quad (5.2)$$

Tables 5.3: Comparison of adjusted and calculated in-situ strength values.

Material	F_{2t}^{is} [MPa]		F_6^{is} [MPa]	
	Adjusted	Calculated	Adjusted	Calculated
IM6/Avimid®K Polymer	64.5	58.6	-	89.1
T300/Fiberite 934	41.3	42.8	-	141.4
AS4/Hercules 3501-6	37.2	89.6	-	131.7
IM7/MTM45-1	50.9	82.4	128.7	67.9

$$F_6^{is} = F_6 \sqrt{\frac{2 t_t}{t_e}} \quad (5.3)$$

where F_{2t}^{is} and F_6^{is} are the unidirectional ply strength (Table 4.1). The comparison is shown in Table 5.3. The variation of F_{2t}^{is}/F_{2t} with the ply thickness is illustrated in Figure 5.8. It can be seen how the in-situ strength value increase exponentially as the ply thickness decreases. The glass transition thickness is $t_t = 0.6$ mm.

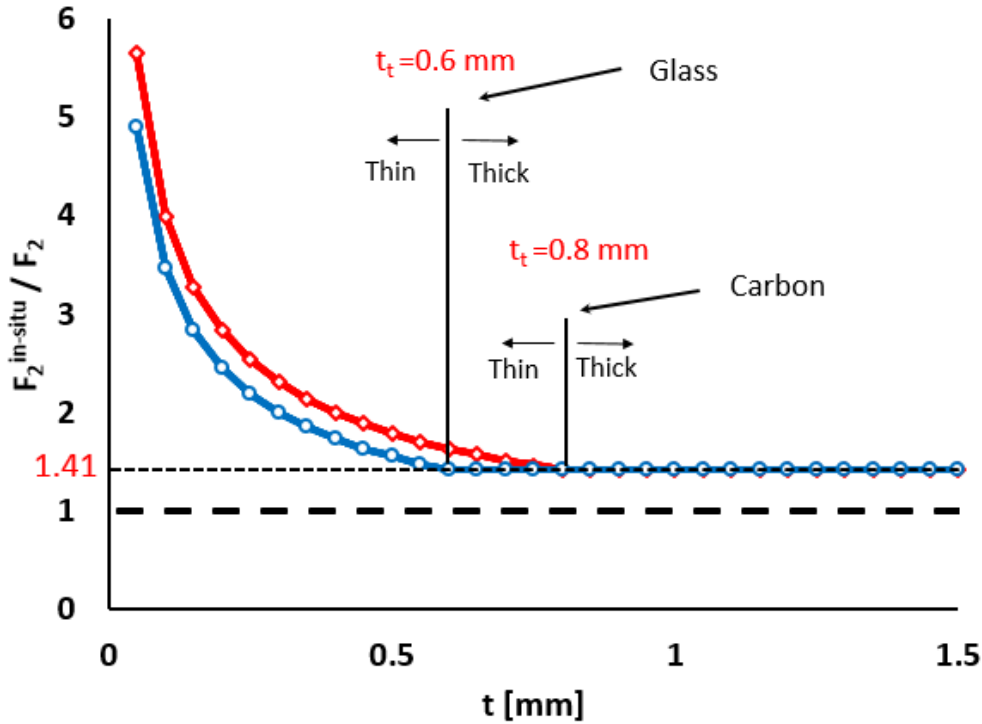


Figure 5.8: Non-dimensional strength predictions vs. ply thickness for transverse tensile strength.

5.4 Convergence and Mesh Sensitivity

The mesh sensitivity of Abaqus PDA is analyzed through h- and p- refinement, namely, mesh refinement and interpolation order (type of element). Since the laminate is subjected to uniform strain, the stress fields should converge independently of mesh refinement. However, the softening material behavior is mesh dependent, as it was mentioned in Section 2.0.2. Such mesh

dependency can be observed in Fig. 5.9 and 5.10. The PDA softening force-displacements assumption (line ABC in Fig. 2.1) yields different results depending on the size of the elements as it can be seen in Figure 5.9. Consequently, the stiffness increases slightly with the number of elements, thus showing mesh dependency as shown in Fig. 5.10.

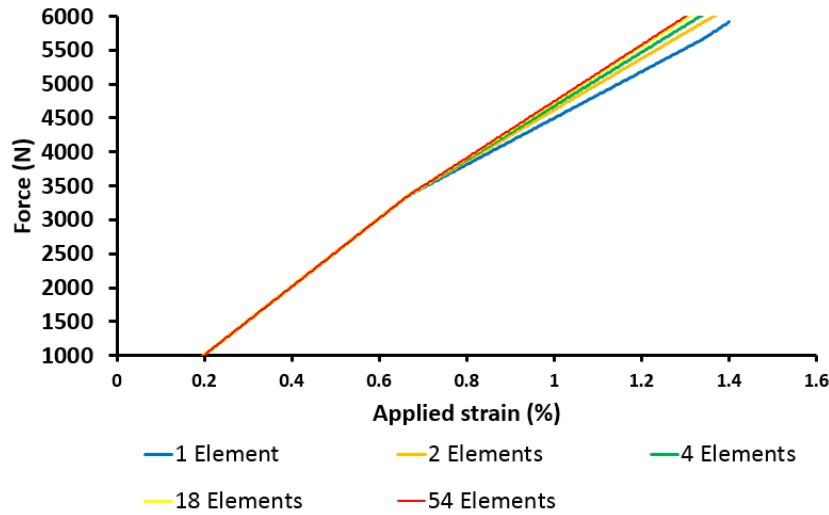


Figure 5.9: Force vs applied strain for laminate 12 as function of number of elements.

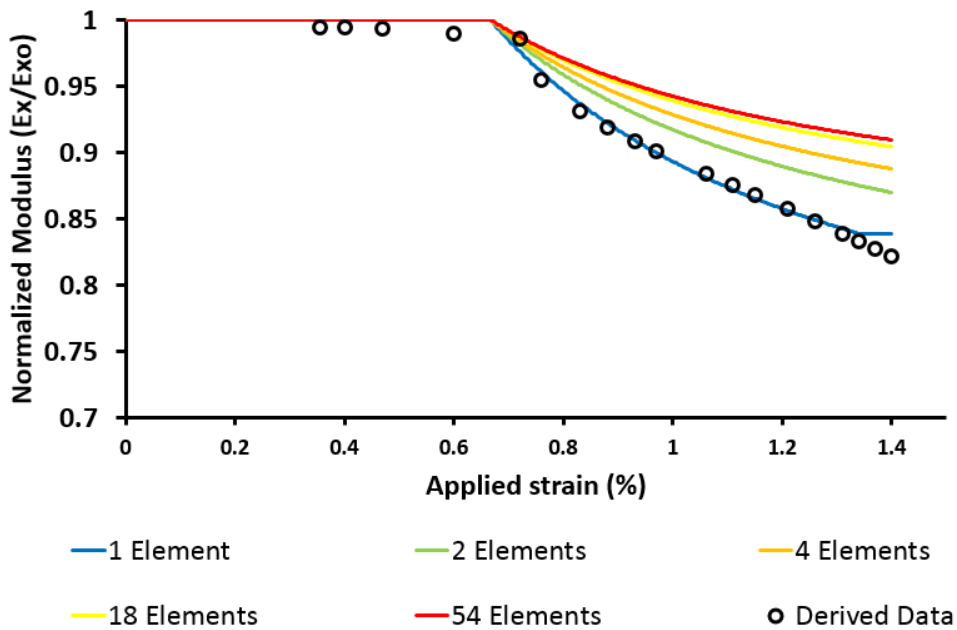


Figure 5.10: PDA model prediction vs. longitudinal modulus from $[0/90_4]_s$ laminate as function of number of elements.

The result of p-refinement, that is comparing S4R (linear interpolation) to S8R (quadratic interpolation) element types is shown in Figure 5.11. Using the properties adjusted with S8R elements into a simulation using S4R elements, sudden degradation of stiffness is observed with S4R. To solve this problem, new parameters are adjusted for S4R, as reported in Table 5.2.

But this is very inconvenient for the user, who would be forced to change material properties depending on what type of element he wishes to use.

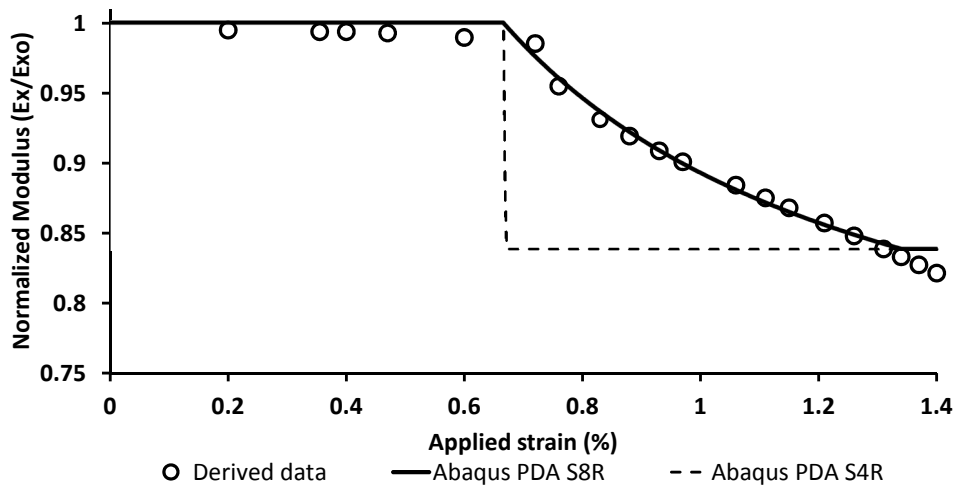


Figure 5.11: PDA model prediction vs longitudinal modulus from laminate $[0/90_4]_s$ IM7/MTM45-1 as function of type of element.

Chapter 6

Conclusions

The material properties F_6^{is} and F_{2t}^{is} are in-situ strength values that cannot be measured by direct experimental methods. The fracture energy G_{mt}^c cannot be measured directly either. Thus, an indirect method is proposed here. For Glass/polymer composites, a straightforward approach can be followed for determination of these properties, by simply adjusting their values to minimize the difference between predicted and measured stiffness of the laminates. For Carbon/polymer composites, a more elaborate method is needed because experimental stiffness reduction data has too low signal/noise ratio. First the properties of the DDM constitutive model are adjusted to minimize the difference between predicted and experimental crack density data. Then, DDM is used to calculate stiffness reduction of those laminates. Then, the calculated stiffness reduction is used to obtain the properties of PDA in the same way as for Glass/polymer composites. Properties can be calculated for all laminates and subsequent predictions match observed behavior accurately. However, some shortcomings of PDA are observed.

PDA assumes that the damage of each lamina can be separated into four uncoupled modes: longitudinal tension, longitudinal compression, transverse tension, and transverse compression. Neither shear nor mode coupling are modeled. In PDA, the softening rate of each mode is controlled by the area under the assumed stress-displacement curve, without interaction among modes. PDA is a local constitutive model in that it does not consider the damage state of neighbor laminas directly into the constitutive equations. As such, the results predicted by PDA are sensitive to the mesh and element-type used as was shown in Fig. 5.9, 5.10, 5.11 which requires adjusting the material properties again or the results would be different. The PDA properties values vary depending on type of element as shown in Table 5.2. These are evidence of mesh sensitivity, also called lack of objectivity even when a *characteristic length* is used in order to avoid as much as possible this mesh dependence.

Furthermore, the lack of a shear-only dissipation energy in PDA makes it difficult for PDA to accurately predict the response of laminates where damage due to in-plane shear dominates the behavior. This is notable for laminates with cracking laminas close to $\pm 45^\circ$ and for laminates

without laminas at 90° (Fig. 5.2 and 5.4). This is a evidence of a bad and erroneous formulation of PDA.

The Abaqus script is the best option with Abaqus because faster execution, simplicity of use, and additional possibilities to extend the scripts usefulness to optimization of other problems. However, MATLAB's optimization functions converge faster to the minimum that those implemented in Scipy.

In conclusion, PDA model is a inaccurate constitutive model which does not follows the formulation of mechanics of material with accuracy. We might conclude that PDA model is computational mechanic model where the lack of accuracy can be solved more or less increasing the mesh to a large number of elements before adjusting the material properties. As shown in Figure 5.10, as we increase the number of element (18 to 54 elements) the deviation becomes smaller avoiding further the mesh dependence but not solution can be given to solve the type of element dependency. Also, the crack density saturation does not match well between DDM and PDA model. The crack saturation is reached early for PDA model in Figures 5.3, 5.6 and 5.7 whose PDA's properties were adjusted to fit with the reduction stiffness slope of generated data (*semi-empirical data*). In order to get the same stiffness slope in PDA, the energy dissipation property G_{mt}^c is slightly small and that is why the crack density saturation is reached early and not more loss of stiffness is produced for major strains. Even, a new damage mode is developed for Figure 5.6 where the fiber tension mode is activated. A big energy property G_{ft}^c was established in order to avoid the sudden laminate breaking.

Appendix A

Abaqus's Python script: example

```

### SCRIPT 1 ###

from abaqus import *
from abaqusConstants import *
from section import *
from part import *
from material import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *

class Model:
    def GetResult(self):
        ## NOW, IT RUNS THE .ODB TO GET THE RESULTS
        viewport = session.Viewport(name='Viewport: 1')
        odb = session.openOdb(name='Job-1.odb')
        viewport.setValues(displayedObject=odb)

        for key in session.xyDataObjects.keys():
            del session.xyDataObjects[key]

        ## GET THE DATA FROM NODES
        session.xyDataListFromField(odb=odb, outputPosition=NODAL, variable=(( 'RF',
            NODAL, ((COMPONENT, 'RF1'), )), ), nodePick=(( 'PLATE-1', 3, ([ '#5 ]', )),
            ), )

        session.xyDataListFromField(odb=odb, outputPosition=NODAL, variable=(( 'U',
            NODAL, ((COMPONENT, 'U1'), )), ), nodePick=(( 'PLATE-1', 1, ([ '#8 ]', )), ),
            )
        xy1 = session.xyDataObjects['U:U1 PI: PLATE-1 N: 4']
        xy2 = session.xyDataObjects['RF:RF1 PI: PLATE-1 N: 1']
        xy3 = session.xyDataObjects['RF:RF1 PI: PLATE-1 N: 3']
        xy4 = session.xyDataObjects['RF:RF1 PI: PLATE-1 N: 8']
        xy5 = combine(xy1, -1*(xy2+xy3+xy4))
        return xy5

    def RunJob(self):
        self.job.submit(consistencyChecking=OFF)
        self.job.waitForCompletion()

    def Setup(self, Gmenergy, F2strenght):
        print "Gmenergy %f F2strenght %f" % (Gmenergy, F2strenght)
        #Creating new databases. Erases all other data.
        mdb = Mdb()

        mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=200.0)
        mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
            point2=(55.0, 10.0))
        mdb.models['Model-1'].Part(dimensionality=THREE_D, name='Plate', type=
            DEFORMABLE_BODY)
        mdb.models['Model-1'].parts['Plate'].BaseShell(sketch=
            mdb.models['Model-1'].sketches['__profile__'])
        del mdb.models['Model-1'].sketches['__profile__']

        ## MATERIAL PROPERTIES
        self.glassEpoxyFiberite = mdb.models['Model-1'].Material(name='glass/epoxy Fiberite HyE
            9082Af')
        self.glassEpoxyFiberite.Elastic(
            table=((

```

```

        44700.0, 12700.0, 12700.0, 0.297, 0.297, 0.41, 5800.0, 5800.0, 4500.0
    ), ),
    type=ENGINEERING_CONSTANTS)
mdb.models['Model-1'].materials['glass/epoxy Fiberite HyE
9082Af'].HashinDamageInitiation(
    table=((1020.0, 620.0, F2strenght, 140.0, 48.5725, 70.0), ))
mdb.models['Model-1'].materials['glass/epoxy Fiberite HyE
9082Af'].hashinDamageInitiation.DamageEvolution(
    table=((1e+30, 1e+30, Gmenergy, 1e+30), ), type=ENERGY)
## SECTION
mdb.models['Model-1'].CompositeShellSection(idealization=NO_IDEALIZATION,
    integrationRule=SIMPSON, layup=(SectionLayer(thickness=0.288,
    material='glass/epoxy Fiberite HyE 9082Af', plyName='K1'), SectionLayer(
    thickness=0.576, orientAngle=90.0,
    material='glass/epoxy Fiberite HyE 9082Af', plyName='K2')), layupName=
    '0_2/90_4', name='Composite layup', poissonDefinition=DEFAULT,
    preIntegrate=OFF, symmetric=True, temperature=GRADIENT, thicknessModulus=
    None, thicknessType=UNIFORM, useDensity=OFF)
## ASSIGN
mdb.models['Model-1'].parts['Plate'].Set(faces=
    mdb.models['Model-1'].parts['Plate'].faces.getSequenceFromMask(['#1'], ),
    ), name='Set-1')
mdb.models['Model-1'].parts['Plate'].SectionAssignment(offset=0.0, offsetField=
    '', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-1'].parts['Plate'].sets['Set-1'], sectionName=
    'Composite layup', thicknessAssignment=FROM_SECTION)
## ASSEMBLY
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=OFF, name='Plate-1',
    part=mdb.models['Model-1'].parts['Plate'])
## STEP
mdb.models['Model-1'].StaticStep(initialInc=0.002, maxNumInc=1500, name='Step-1',
    noStop=OFF, previous='Initial', timeIncrementationMethod=FIXED)
## FIELD OUTPUT
mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(
    sectionPoints=(1, 2, 3, 4, 5, 6), variables=('S', 'U', 'RF', 'DAMAGET',
    'DAMAGEMT', 'HSNMTCRT'))
## B.C AND DISPLACEMENTS. ALSO, IT TAKES THE SET FOR EACH EDGE
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'].edges.getSequenceFromMask(
    ('[#8]', ), ), name='Set-1')
mdb.models['Model-1'].XsymmBC(createStepName='Initial', localCsys=None, name=
    'BC-1', region=mdb.models['Model-1'].rootAssembly.sets['Set-1'])
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'].edges.getSequenceFromMask(
    ('[#1]', ), ), name='Set-2')
mdb.models['Model-1'].YsymmBC(createStepName='Initial', localCsys=None, name=
    'BC-2', region=mdb.models['Model-1'].rootAssembly.sets['Set-2'])
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'].edges.getSequenceFromMask(
    ('[#2]', ), ), name='Set-3')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-3', region=mdb.models['Model-1'].rootAssembly.sets['Set-3'], u1=0.825,
    u2=UNSET, u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=UNSET)
## MESH
# SEED
mdb.models['Model-1'].rootAssembly.seedPartInstance(deviationFactor=0.1,
    minSizeFactor=0.1, regions=(
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'], ), size=55.0)
# CONTROL MESH
mdb.models['Model-1'].rootAssembly.setMeshControls(elemShape=QUAD, regions=
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'].faces.getSequenceFromMask(
    ('[#1]', ), ), technique=STRUCTURED)

```

```
# ELEMENT TYPE
mdb.models['Model-1'].rootAssembly.setElementType(elemTypes=(ElemType(
    elemCode=S8R, elemLibrary=STANDARD), ElemType(elemCode=STRI65,
    elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'].faces.getSequenceFromMask(
        ('[#1 ]', ), ), ))
mdb.models['Model-1'].rootAssembly.generateMesh(regions=(
    mdb.models['Model-1'].rootAssembly.instances['Plate-1'], ))
# CREATE A JOB
self.job = mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-1', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Job-1', nodalOutputPrecision=SINGLE,
    numCpus=1, numGPUs=0, queue=None, resultsFormat=ODB, scratch='', type=
    ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
```

```

### SCRIPT 2 ###

import Model
import math
#[Gmc,F2t,E (MPa),Area (mm^2), l_o (mm),Tol (mm)]
def Iteration(x, *args):
    Gmenergy = x[0]
    F2strenght = x[1]
    E = args[0]
    Area = args[1]
    l_o = args[2]

    model = Model.Model()

    model.Setup(Gmenergy, F2strenght)

    model.RunJob()
    result = model.GetResult()

    displacements = [element[0] for element in result[1:]]
    forces = [element[1] for element in result[1:]]

    strains = [d / l_o for d in displacements]
    stresses = [f / Area for f in forces]

    stiffness = [stress / strain / E for (stress, strain) in zip(stresses, strains)]

    strains = [strain * 100.0 for strain in strains]

    A = dict(zip(strains, stiffness))

    B = {
        0.35909      : 1,
        0.364866    : 1,
        0.399523    : 1,
        0.509413    : 0.984729,
        0.578784    : 0.978818,
        0.590251    : 0.987685,
        0.711778    : 0.964039,
        0.764191    : 0.919704,
        0.769995    : 0.916749,
        0.787295    : 0.919704,
        0.967324    : 0.819212,
        0.97882     : 0.825123,
        0.996091    : 0.831034,
        1.01353     : 0.819212,
        1.18127     : 0.795567,
        1.19288     : 0.789655,
        1.21618     : 0.768966,
        1.23343     : 0.777833,
        1.378       : 0.760099,
        1.40125     : 0.74532,
        1.41277     : 0.748276,
        1.47645     : 0.733498
    }
    error = 0
    for (experimentalStrain, experimentalStiffness) in B.items():
        (_, correspondingModelStiffness) = min(A.items(), key=lambda (v, _): abs(v -
            experimentalStrain))
        #print str(experimentalStiffness) + " " + str(correspondingModelStiffness)
        error += pow(experimentalStiffness - correspondingModelStiffness, 2)

    error = (1.0/len(B)) * math.sqrt(error);
    return error

```



```
### SCRIPT 3 ###

import numpy as np

from numpy import array, asarray, float64, int32, zeros

from scipy.optimize import fmin

from PyFilescripttotal import Iteration

#Gmc = 3 # Toughness fracture (initial value)
#F2t = 31 # Transverse in-situ strength (initial value)

#lb = [1.5, 20] # Initial bound constraint for Gmc and F2t respectively
#ub = [30.5, 141] # Final bound constraint for Gmc and F2t respectively

x0 = array([3, 31]) # Initial vector
#mybounds = [(1.5,20), (30.5,141)] # Initial bound constraint vector

E = 23540 # Initial longitudinal modulus of composite
Area = 17.28 # Total area of composite
l_o = 55 # Initial length of composite

xopt,fopt,iter,funcalls, warnflag,allvecs = fmin(Iteration, x0, args = (E, Area, l_o),
xtol=1e-4,ftol=1e-4, maxiter=None, maxfun=None, full_output=True, retall=True)

print(xopt)
print(fopt)
print(funcalls)
print(allvecs)
```

Bibliography

- [1] E. J. Barbero, *Finite Element Analysis of Composite Materials Using Abaqus*, CRC Press, 2013.
- [2] E. Barbero, G. Sgambitterra, A. Adumitroaie, X. Martinez, A discrete constitutive model for transverse and shear damage of symmetric laminates with arbitrary stacking sequence, *Composite Structures*.
- [3] A. Abad Blazquez, M. Herraiz Matesanz, C. Navarro Ugena, , E. Barbero, Acoustic emission characterization of intralaminar damage in composite laminates, Master's thesis, Carlos III University, Spain (2013).
- [4] S. Liu, J. A. Nairn, Formation and propagation of matrix microcracks in cross-ply laminates, *J. Reinf. Plas. Compos* (1992) 158–178.
- [5] E. Barbero, F. Cosso, Determination of material parameters for discrete damage mechanics analysis of carbon-epoxy laminates, *Composites Part B* (2014) 638–646.
- [6] E. Barbero, *Introduction to Composite Materials Design*, second edition Edition, CRC Press, 2011.
- [7] R. J. Nuismer, S. C. Tan, Constitutive relations of a cracked composite lamina, *Journal of Composite Materials* 22 (1988) 306–321.
- [8] S. C. Tan, R. J. Nuismer, A theory for progressive matrix cracking in composite laminates, *Journal of Composite Materials* 23 (1989) 1029–1047.
- [9] S. C. Tan, R. J. Nuismer, A continuum damage model for transverse matrix cracking in laminated fibre-reinforced composites, *Physical and Engineering Sciences* 356 (1998) 2379–2412.
- [10] S. Kim, J. A. Nairn, Fracture mechanics analysis of coating/substrate systems. part ii: Experiments in bending, *Engineering Fracture Mechanics* (200) 1–25.
- [11] J. A. Nairn., Finite fracture mechanics of matrix microcracking in composites, *Application of Fracture Mechanics to Polymers, Adhesives and Composites* (2004) 207–212.

- [12] S. H. Lim, S. Li, Energy release rates for transverse cracking and delaminations induced by transverse cracks in laminated composites, *Composites Part A* 36(11) (2005) 1467–1746.
- [13] D. T. G. Katerelos, J. Varna, C. Galiotis, Energy criterion for modeling damage evolution in cross-ply composite laminates, *Composites Science and Technology* 68 (2008) 2318–2324.
- [14] D. T. G. Katerelos, M. Kashtalyan, C. Soutis, C. Galiotis, Matrix cracking in polymeric composites laminates: Modeling and experiments, *Composites Science and Technology* 68 (2008) 2310–2317.
- [15] J. A. Mayugo, P. P. Camanho, P. Maimi, C. G. Davila, Analytical modeling of transverse matrix cracking of [ÅŒÿ/90n] composite laminates under multiaxial loading, *Mechanics of Advanced Materials and Structures* 17 (2010) 237–245.
- [16] G. V. T. . A. Software, Alpha star corporation.
URL <http://www.ascgenoa.com>
- [17] H. C. M. A. Software.
- [18] P. Camanho, C. Davila, Mixed-mode decohesion finite elements for the simulation of delamination in composite materials, *NASA/TM* (2002) 1–37.
- [19] A. Matzenmiller, J. Lubliner, R. Taylor, A constitutive model for anisotropic damage in fiber-composites, *Mechanics of Materials* 20 (1995) 125–152.
- [20] Z. Hashin, A. Rotem, A fatigue failure criterion for fiber-reinforced materials, *Journal of Composite Materials* (1973) 448–464.
- [21] Abaqus online Documentation, <http://129.97.46.200:2080/v6.14/books/hhp/default.htm>.
- [22] E. Barbero, D. Cortes, A mechanistic model for transverse damage initiation, evolution, and stiffness reduction in laminated composites, *Composites Part B* (2010) 124–132.
- [23] E. Barbero, *Finite element analysis of composite materials*, boca raton Edition, CRC Press, 2008.
- [24] J. Nairn, On the use of shear-lag methods for analysis of stress transfer in unidirectional composites, *Journal of Mechanics of Materials* 26 (1977) 63–80.
- [25] E. Kreyszig, *Advanced Engineering Mathematics*, 10th Edition, Wiley, 2011.
- [26] J. Rebiere, D. Gamby, A decomposition of the strain energy release rate associated with the initiation of transverse cracking, longitudinal cracking and delamination in cross-ply laminates, *Composite Structures* 84 (2008) 186–197.
- [27] H. Tada, P. Paris, G. Irwin, *The stress analysis of cracks handbook*, ASME Press, 2000.

- [28] H. Hahn, A mixed-mode fracture criterion for composite materials, *Composites Technology* 5 (1983) 26–29.
- [29] A. F. Bower, *Mechanics of solids*, CRC Press, 2010.
- [30] A. Adumitroaie, E. Barbero, Intralaminar damage model for laminates subjected to membrane and flexural deformations, *Mechanics of Advanced Materials and Structures*.
- [31] C. Sing, F. A. Cosso, R. Roman, T. L. Weadon., Determination of material parameters for abaqus progressive damage analysis of e-glass epoxy laminates, *Composites Part B:Engineering* 46 (2013) 211–220.
- [32] J. Varna, R. Joffe, N. Akshantala, R. Talreja, Damage in composite laminates with off-axis plies, *Composites Science and Technology* (1999) 2139–2148.
- [33] J. Varna, R. Joffe, R. Talreja, A synergistic damage mechanics analysis of transverse cracking in $[\pm 45/90]_s$ laminates, *Composites Science and Technology* (2001) 657–665.
- [34] J. A. Nairn, The strain energy release rate of composite microcracking: A variational approach, *Journal of Composite Materials* 23 (1989) 1106–1129.
- [35] J. A. Nairn, H. Shoufeng, J. Bark, A critical evolution of theories for predicting microcracking in composite laminates, *Journal of Material Science* 28 (1993) 5099–5111.
- [36] E. Barbero, R. Talreja, Evolution of ply cracks in multidirectional composite laminates, *Journal of Solids and Structures* 47 (2010) 1338–1349.
- [37] E. Barbero, J. Cabrera Barbero, C. Navarro Ugena, Analytical solution for plane stress/strain deformation of laminates with matrix cracks, *Composite Structures* (2015) 621–632.
- [38] E. Barbero, J. Cabrera Barbero, Analytical solution for bending of laminated composites with matrix cracks, *Composite Structures* (2016) 140–155.