

Determination of Material Properties for Progressive Damage Analysis of Carbon/Epoxy Laminates

Ever J. Barbero¹ and Javier Cabrera Barbero²

Abstract

Progressive Damage Analysis is a constitutive model available in Abaqus(TM) to predict damage initiation and evolution in laminated composite materials but no standards are available to obtain the required material properties. A novel methodology is proposed to determine the material properties for Carbon/Polymer composites, relying on crack-density data. Since PDA does not predict crack density, direct correlation of data with PDA predictions is not possible. Therefore, another constitutive model is used to estimate modulus-reduction as a function of measured crack density. Then, PDA material properties are determined using the calculated modulus-reduction data. Applicability to several material systems is presented.

Keywords

Polymer-matrix composites; Matrix damage; Damage mechanics; Parameter identification; Mesh dependency; Objectivity.

1 Introduction

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, modulus-reduction, and failure. Simulation techniques are able to predict damage initiation and evolution as a function of service conditions. For instance, the predictive capability of Progressive Damage Analysis (PDA) relies on material properties to characterize the ability of the composite to delay onset and progression of damage [1–4]. Although data and standard methods exist to measure elastic moduli (e.g., ASTM D3039), data and methods do not exist for damage-related properties. Experimental data displaying macroscopic effects of damage (e.g., modulus-reduction) exist for Glass/Polymer composites but not for Carbon/Polymer composites due to the difficulty for measuring modulus-reduction for the later. However, data and methods to measure crack density vs. applied stress or applied strain exist for Carbon/Polymer material systems [5–8] and they are sufficiently standardized to be used for other material systems. Unidirectional-lamina strength data and standard test methods to measure them also exist, but once the lamina is embedded into a laminate, in-situ correction is needed for the transverse tensile and in-plane shear strength [9, § 7.2.1].

¹Professor, West Virginia University, Morgantown, USA. Corresponding author.

²Graduate Research Assistant, West Virginia University, Morgantown, USA.

The final publication is available at <http://dx.doi.org/10.1080/15376494.2018.1430281>

Such correction depends on the *intralaminar* energy release rate in modes I and II, but no standard tests methods exist to measure the later.

The purpose of this study is to develop a methodology 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 are compared to a broad set of Carbon/Epoxy laminates with different laminate stacking sequences. Initially, the proposed method relies on availability of modulus-reduction vs. applied load or strain. As it was pointed out in [1], 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 may go unnoticed during modulus-reduction measurements. The sensitivity of modulus-reduction to damage varies depending on the laminate stacking sequence (LSS), as reported in [10]. Even for laminates having a high proportion of 90° laminas, such as $[0/90_2]_S$ to $[0/90_4]_S$, the modulus-reduction is in the range 0–5% to 0–15% approximately, making it difficult to measure accurately, specially at early stages of damage. For other LSS, such as quasi-isotropic, modulus-reduction is even less, because off-axis laminas say (e.g., ± 45 , ± 60 , etc.) damage less than transverse laminas (90°) and longitudinal plies (0°) often do not damage at all. For this reason, modulus-reduction data for Carbon/Polymer composites is difficult to find in the literature.

A more direct measure of damage, i.e., crack density (cracks/mm), is often reported in the literature [5–8, 11–14], 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 modulus-reduction in terms of available crack-density data using an intermediate damage mechanics model [15] that uses crack density data to predict modulus-reduction. Then, the derived stiffness data is used to obtain the material properties needed for using PDA in Abaqus.

2 Progressive Damage Analysis (PDA)

The Abaqus PDA model is a generalization of an interlaminar decohesion model [2, 3]. It assumes linearly elastic behavior of the undamaged material and it is used in combination with Hashin's damage initiation criteria [16].

2.1 Damage initiation

Abaqus assumes linear elastic behavior of the undamaged material until damage initiates. Damage initiation is predicted using Hashin's failure criterion for each mode (fiber tension, fiber compression, matrix tension, matrix compression). The four different damage initiation mechanisms, which can be coupled or uncoupled, are defined as follows.

I. Fiber tension ($\sigma_{11} \geq 0$) is controlled by the longitudinal tensile strength of the unidirectional lamina F_{1t} and a correction to take into account fiber shear failure, as follows

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

II. Fiber compression ($\sigma_{11} < 0$) is controlled by the longitudinal compressive strength of the unidirectional lamina F_{1c} [17] as follows

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

III. Matrix tension and/or shear ($\sigma_{22} \geq 0$), which is dominated by matrix cracks driven by a combination of transverse tensile and in-plane shear stress, as follows

$$I_m^t = \left(\frac{\sigma_{22}}{F_{2t}} \right)^2 + \left(\frac{\sigma_{12}}{F_6} \right)^2 \quad (3)$$

IV. Matrix compression ($\sigma_{22} < 0$), which results in shear-compression failure of the matrix

$$I_m^c = \left(\frac{\sigma_{22}}{2F_4} \right)^2 + \left[\left(\frac{F_{2c}}{2F_4} \right)^2 - 1 \right] \frac{\sigma_{22}}{F_{2c}} + \left(\frac{\sigma_{12}}{F_6} \right)^2 \quad (4)$$

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 determines the contribution of shear stress to the fiber tension mode. To obtain the model proposed by Hashin and Rotem [16] 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 the 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 (F_{2t} , F_6) 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 behaves as if it were stronger than the unidirectional lamina [9, 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 lamina strength and thickness values [9, 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 tensor of the lamina [4] as follows

$$\sigma = C : \epsilon \quad (5)$$

where σ is the apparent stress, ϵ the strain, C the damaged stiffness matrix, and $:$ denotes a tensor double contraction. In addition, the damaged stiffness matrix can be written in terms of damage variables as follows

$$C = \begin{bmatrix} (1 - d_f)E_1/\Delta & (1 - d_f)(1 - d_m)\nu_{21}E_1/\Delta & 0 \\ (1 - d_f)(1 - d_m)\nu_{12}E_2/\Delta & (1 - d_m)E_2/\Delta & 0 \\ 0 & 0 & (1 - d_s)G_{12} \end{bmatrix} \quad (6)$$

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

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

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 (1-4). At any instant of time, each variable is updated according whether it is in tension or compression as follows

$$d_f = \begin{cases} d_f^t & \text{if } \sigma_{11} \geq 0 \\ d_f^c & \text{if } \sigma_{11} < 0 \end{cases} \quad (7)$$

and

$$d_m = \begin{cases} d_m^t & \text{if } \sigma_{22} \geq 0 \\ d_m^c & \text{if } \sigma_{22} < 0 \end{cases} \quad (8)$$

Abaqus PDA uses the model proposed by Matzenmiller et al. [3] to compute the degradation of stiffness matrix coefficients. The equations (1-4) are then used as damage evolution criteria by substituting effective stresses $\tilde{\sigma}$ for Cauchy stress σ in (1-4). The relationship between the effective stress $\tilde{\sigma}$ and the Cauchy stress σ is given by [18]

$$\tilde{\sigma} = M^{-1} : \sigma \quad (9)$$

and the damage effect tensor in Voigt notation is given as

$$M^{-1} = \begin{bmatrix} (1 - d_f)^{-1} & 0 & 0 \\ 0 & (1 - d_m)^{-1} & 0 \\ 0 & 0 & (1 - d_s)^{-1} \end{bmatrix} \quad (10)$$

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 also in the criteria for damage initiation of other modes through (1-4). When $d_f = 0$ and $d_m = d_s = 1$, equation (10) represents the ply discount method.

2.2 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 values 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 area of the triangle OAC in Figure 1 represents the critical energy-dissipation for mode III (3).

Usually the constitutive model is expressed in terms of stress-strain, but when the material exhibits strain-softening behavior, as shown in Figure 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 to model it. In order to alleviate the mesh dependency, PDA uses a *characteristic length* (L_c) to transform the PDA constitutive model from *stress-strain* to *strain-displacement* by computing the product $\delta = \epsilon L_c$. 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 variables. Unfortunately, the damage variables cannot be measured experimentally, but only inferred from modulus-reduction.

The evolution of damage variables is governed by an equivalent displacement δ^{eq} shown in Figure 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 [4]:

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

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

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

$$\begin{aligned}\delta_{fc}^{eq} &= L^c \langle -\epsilon_{11} \rangle \\ \sigma_{fc}^{eq} &= \frac{\langle -\sigma_{11} \rangle \langle -\epsilon_{11} \rangle}{\delta_{fc}^{eq} / L^c}\end{aligned}\quad (12)$$

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

$$\begin{aligned}\delta_{mt}^{eq} &= L^c \sqrt{\langle -\epsilon_{22} \rangle^2 + \epsilon_{12}^2} \\ \sigma_{mt}^{eq} &= \frac{\langle \sigma_{22} \rangle \langle \epsilon_{22} \rangle + \sigma_{12} \epsilon_{12}}{\delta_{mt}^{eq} / L^c}\end{aligned}\quad (13)$$

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

$$\begin{aligned}\delta_{mc}^{eq} &= L^c \sqrt{\langle -\epsilon_{22} \rangle^2 + \epsilon_{12}^2} \\ \sigma_{mc}^{eq} &= \frac{\langle -\sigma_{22} \rangle \langle -\epsilon_{22} \rangle + \sigma_{12} \epsilon_{12}}{\delta_{mc}^{eq} / L^c}\end{aligned}\quad (14)$$

where $\langle \eta \rangle$ 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 1 as follows

$$d = \frac{\delta_c^{eq}(\delta_c^{eq} - \delta_o^{eq})}{\delta_c^{eq}(\delta_c^{eq} - \delta_o^{eq})}\quad (15)$$

where δ_c^{eq} is the maximum value of δ^{eq} at point C in Fig 1, for each mode.

In PDA, a material point is initially stressed and strained along the linear elastic line OA in Fig. 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 [9, section 7.2.1]. In-situ values can be obtained (adjusted) by matching PDA model predictions to the initiation of laminate modulus-reduction data. 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 (Section 4).

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 the evolution of laminate modulus-reduction data (Section 4).

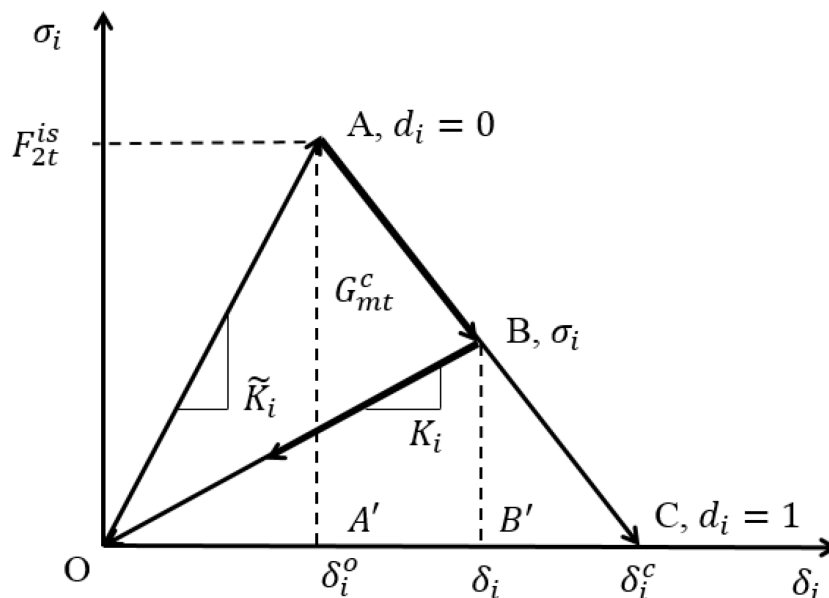


Figure 1: Energy-dissipation property for one damage mode.

3 Modulus-reduction Data

In this section, experimental data of crack density vs. applied strain or applied stress is used to calculate the damage material properties for the discrete damage mechanics DDM model [15] as well as to generate the modulus-reduction data that it is needed to adjust the PDA material properties introduced in Section 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. Often, experimental errors are of the same order of magnitude of the modulus-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 [5–8].

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 used in the experiments were subjected to fiber or matrix mode compression (equations (2) and (4)). 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 modulus-reduction data using another constitutive model that calculates crack density and thus can be compared to crack density data. Once the modulus-reduction data has been generated, it is then used to adjust the material properties for PDA through an optimization process (Section 4).

A discrete damage mechanics model (DDM) [15] was chosen to generate modulus-reduction data. DDM is implemented as a user general section (UGENS) in Abaqus [18]. The material properties of the discrete damage mechanics (DDM) constitutive equation can be adjusted by comparing DDM predictions of crack density with experimental crack-density data, which is either available [5–7] or can be obtained by established methods [8]. The material property determination for DDM is performed by executing a Python optimization script 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 modulus-reduction, via DDM, as required by

the optimization algorithm. The error while adjusting DDM properties is calculated as

$$Error = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\lambda|_{\epsilon=\epsilon_i}^{Abaqus} - \lambda|_{\epsilon=\epsilon_i}^{Experimental} \right)^2} \quad (16)$$

where λ^{Abaqus} and $\lambda^{Experimental}$ are the calculated and measured values of crack density for each value of applied strain ϵ_i , respectively, and $i = 1 \dots N$ correspond to the available experimental data points.

DDM requires just two properties, the true energy release rates (ERR) in modes I (crack opening) G_I^c and II (crack shear) G_{II}^c . Note that the fracture energy G_{mt}^c in PDA is not true 3D ERR and thus it is not numerically equal to G_I^c or G_{II}^c 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 (11-14).

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_I^c (crack opening mode I). On the other hand, when laminates with $\pm\theta^\circ$ laminas are subjected to uniaxial extension, both traction and shear may appear, so damage initiation and evolution are controlled by both G_I^c and G_{II}^c . DDM not only calculates the crack density but also the modulus-reduction of the laminate, thus providing the data needed for adjusting the properties needed for PDA.

Once the modulus-reduction data E/\tilde{E} has been generated, the PDA properties can be adjusted by minimizing the PDA error as follows

$$Error = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\frac{E}{\tilde{E}} \Big|_{\epsilon=\epsilon_i}^{Abaqus} - \frac{E}{\tilde{E}} \Big|_{\epsilon=\epsilon_i}^{Data} \right)^2} \quad (17)$$

where E , \tilde{E} , are the damaged and undamaged modulus for applied strain ϵ_i , respectively, and N is the number of data points.

The methodology proposed in this section is applied next to generate modulus-reduction data E/\tilde{E} for several material systems. Such data is then used in Section 4 to adjust the material properties required by PDA.

All the laminates considered in this 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 to one end of the specimen. The dimensions of the specimens are 12 mm wide with a free length of 150 mm, except when noted otherwise. Furthermore, the DDM predictions are insensitive mesh density and type of element used, namely linear S4R or quadratic S8R, so a one-element mesh with either type of element can be used. The ply material properties are listed in Table 1.

3.1 IM6/Avimid®K Polymer

Crack density λ vs applied stress σ_x measured experimentally [5] are used to adjust the DDM properties G_I^c , G_{II}^c and then generate modulus-reduction data for IM6/Avimid-K laminates. Laminate #2 in Table 2 was used to adjust G_I^c because it shows the strongest mode I fracture behavior of the group [19]. The calculated value G_I^c is reported in Table 3. Note that there are no laminates subjected to shear reported in [5], so G_{II}^c cannot be calculated from the data available. Comparison between crack density predicted by DDM and experimental data is shown in Figure 2. The modulus-reduction generated by DDM for $[0/90_3]_s$ laminate IM6/Avimid-K is reported in Figure

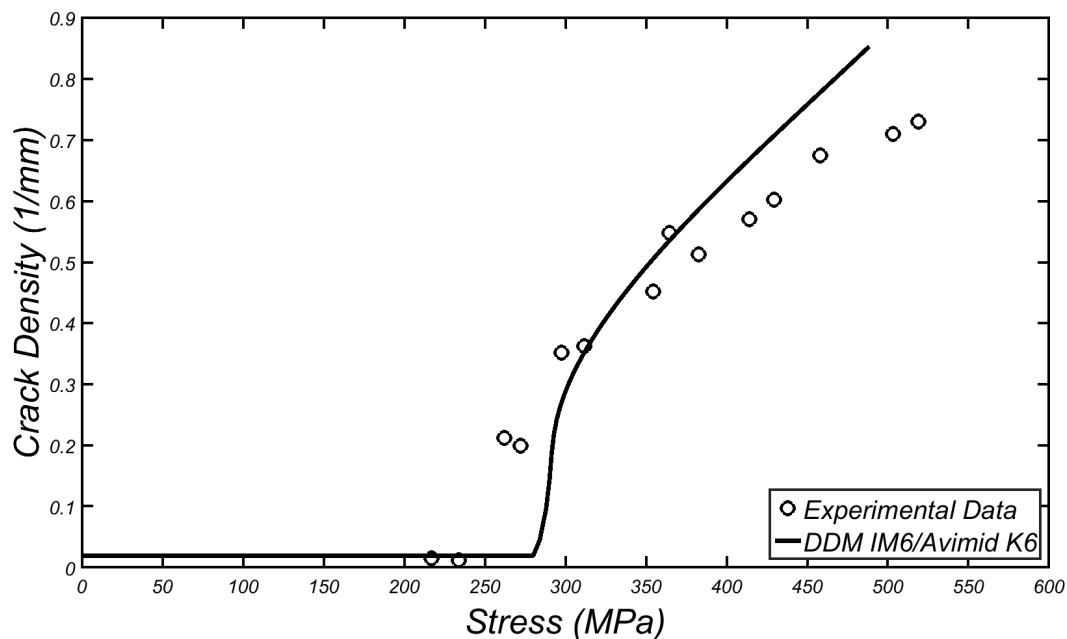


Figure 2: DDM model prediction and crack density data in $[0/90_3]_s$ laminate IM6/Avimid®K Polymer.

3, denoted as "generated modulus-reduction data". This data will be used in Section 4 to calculate the material properties required by PDA.

3.2 T300/Fiberite 934

Crack density λ vs applied stress σ_x measured experimentally [5] (Figure 4) are used to adjust the DDM properties G_I^c, G_{II}^c and then generate modulus-reduction data for T300/934 laminates. Laminate #9 in Table 2 was used to adjust G_I^c because it shows the strongest mode I fracture behavior of the group [19]. The calculated value G_I^c is reported in Table 3. Note that there are no laminates subjected to shear reported in [5], so G_{II}^c cannot be calculated from the data available.

3.3 AS4/Hercules 3501-6

Crack density λ vs applied stress σ_x measured experimentally [5] are used to adjust the DDM properties G_I^c, G_{II}^c and then generate modulus-reduction data for AS4/3501-6 laminates. Laminate #10 in Table 2 was used to adjust G_I^c because it shows the strongest mode I fracture behavior of the group [19]. The calculated value G_I^c is reported in Table 3. Note that there are no laminates subjected to shear reported in [5], so G_{II}^c cannot be calculated from the data available.

Comparison between crack density predicted by DDM and experimental data is shown in Figure 5. Simultaneously, the modulus-reduction generated by DDM for $[0/90_4]_s$ laminate AS4/3501-6 is reported in Figure 3. This data will be used in Section 4 to calculate the material properties required by PDA.

In Figure 5, the effect of crack saturation is likely responsible for the difference because crack saturation is not accounted for the DDM model. As crack density approaches one crack per mm ($\lambda = 1.0/\text{mm}$), cracks become too close to each other, closer than the shear lag distance [10]

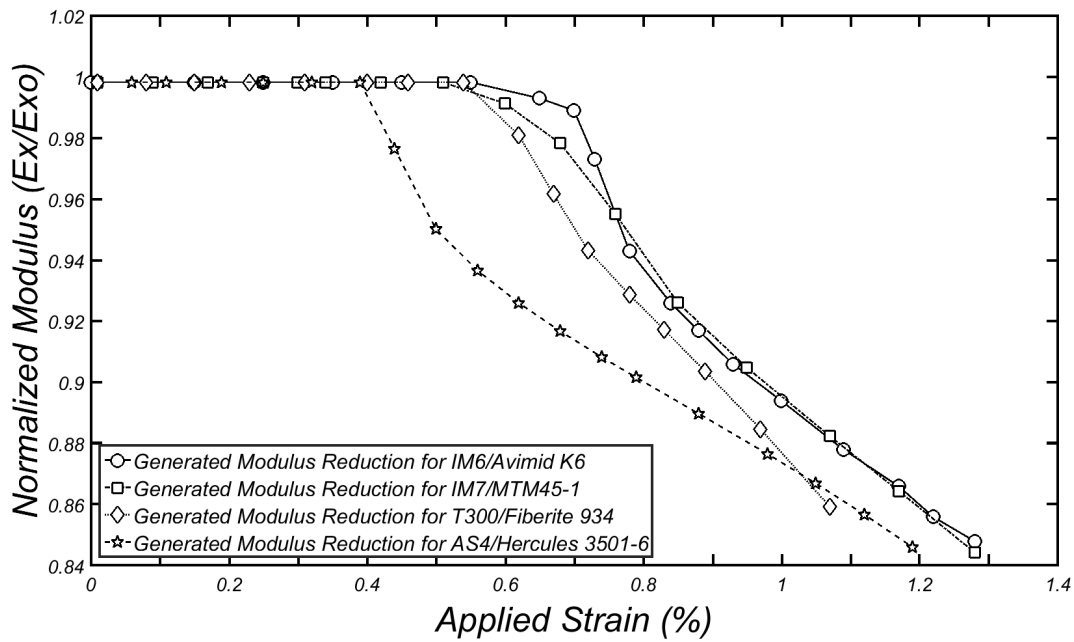


Figure 3: Modulus reduction generated by DDM for several material systems: $[0/90_3]_s$ laminate IM6/Avimid®K Polymer, $[0_2/90_4]_s$ laminate T300/Fiberite 934, $[0/90_2]_s$ laminate AS4/Hercules 3501-6 and $[0/90_4]_s$ laminate IM7/MTM45-1.

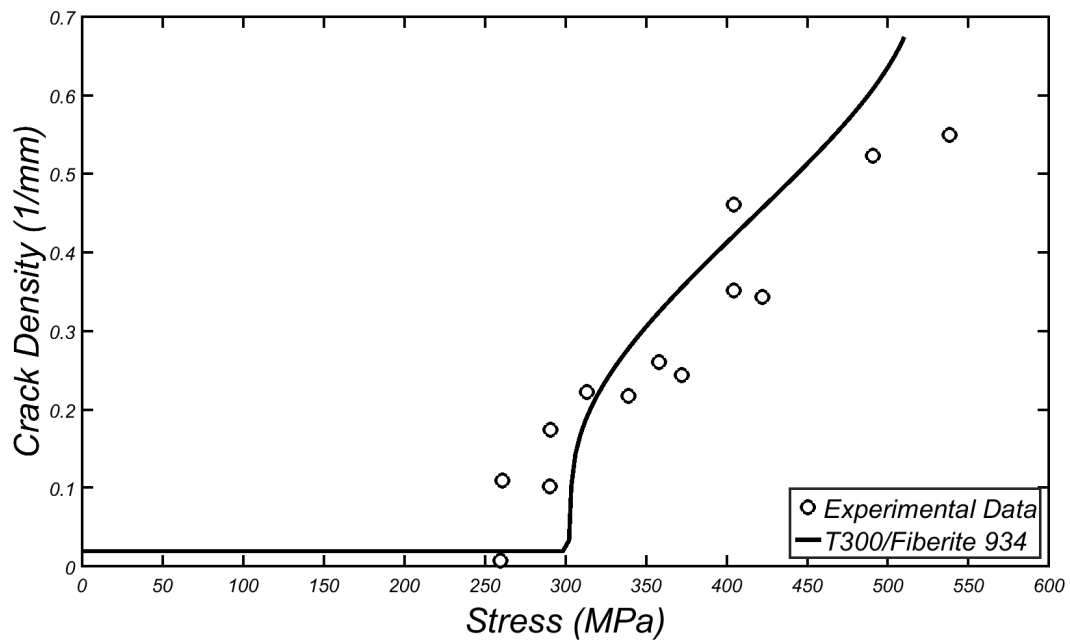


Figure 4: DDM model prediction and crack density data in $[0_2/90_4]_s$ laminate T300/Fiberite 934.

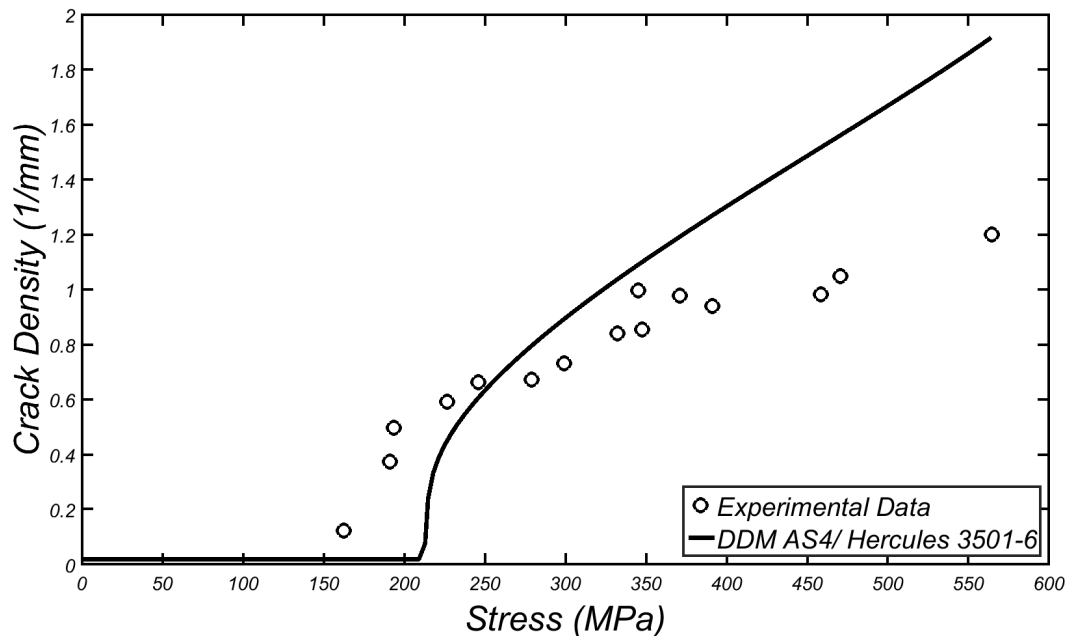


Figure 5: DDM model prediction and crack density data in $[0/90_2]_s$ laminate AS4/Hercules 3501-6.

required for the intralaminar shear to transfer the stress from one crack to the next. This effects slows down the experimental crack density vs. strain rate with respect to the predicted one.

3.4 IM7/MTM45-1

Crack density λ vs applied strain ϵ_x measured experimentally [8] are used to adjust the DDM properties G_I^c, G_{II}^c and then generate modulus-reduction data for IM7/MTM45-1 laminates. Laminate #12 in Table 2 was used to adjust G_I^c because it shows the strongest mode I fracture behavior of the group. In the same way, laminate #14 was chosen because it shows a strong shear component, thus the strongest mode II fracture behavior. The DDM properties that yield the best match to crack density data are reported in Table 3.

Comparison between crack density predicted by DDM and experimental data [8] is shown in Figure 6 and 7 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 1. Modulus-reduction generated by DDM for $[0/90_4]_s$ laminate IM7/MTM45-1 is illustrated in Figure 3, denoted as "generated modulus-reduction data". This data will be used in Section 4 to calculate the material properties required by PDA.

In Figure 6 for $[0/90_4]_s$, discrepancies are observed after 1.13 % strain. At that strain, it is likely that fiber failures start to weaken the laminate, thus accelerating matrix cracking beyond that predicted by the model, while the crack density is still too low for the results to suffer from the effects of crack saturation. Since the laminate $[0/\pm 55/0/\pm 55/0]$ in Figure 7 has no 90° laminas, but instead has $\pm 55^\circ$ laminas that are less susceptible to cracking, discrepancies are not observed until about 1.38 % strain. At that strain, it is likely that fiber failures start to accelerate matrix cracking beyond that predicted by the model, while the crack density is still too low for the results to suffer from the effects of crack saturation.

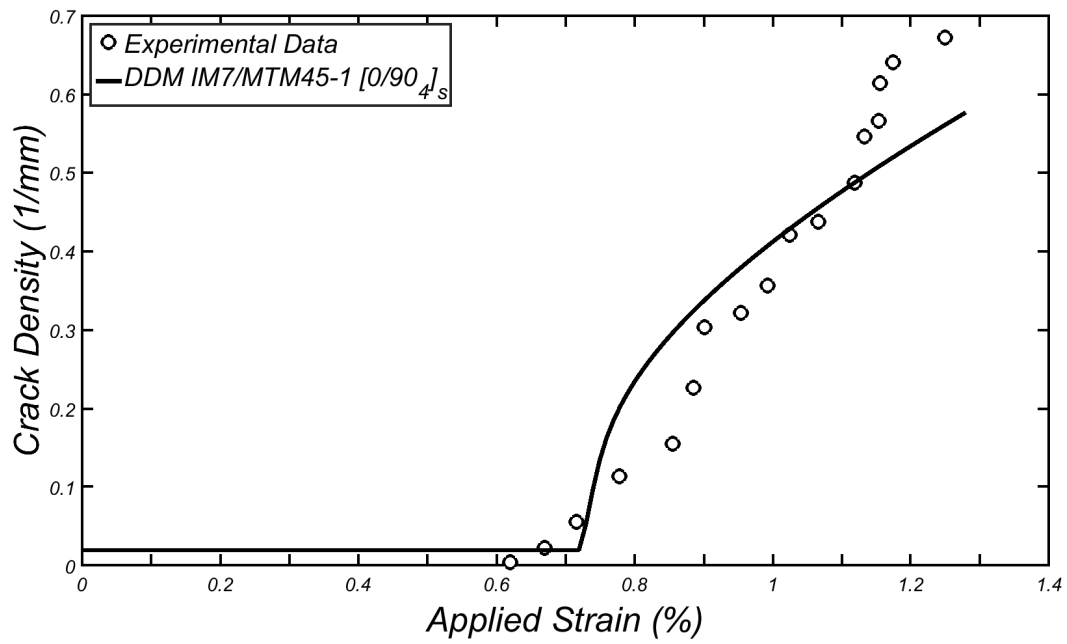


Figure 6: DDM model prediction and crack density data in $[0/90_4]_s$ laminate IM7/MTM45-1.

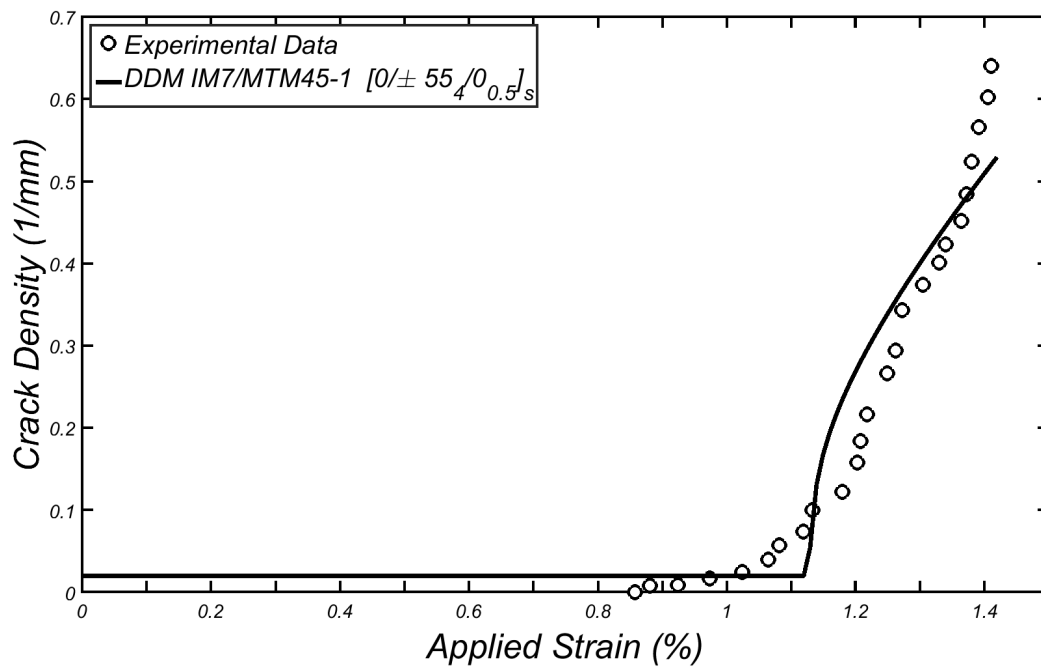


Figure 7: DDM model prediction and crack density data in $[0/\pm 55_4/0_{0.5}]_s$ laminate IM7/MTM45-1.

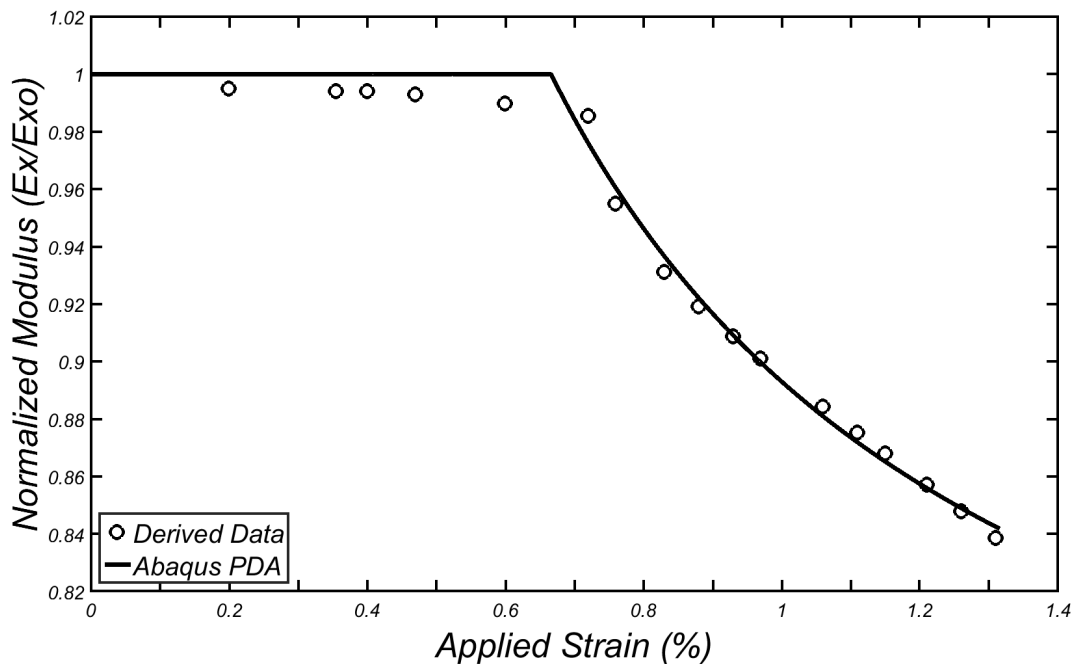


Figure 8: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_4]_s$ IM7/MTM45-1.

4 Adjust PDA Material Properties

In this section, the modulus-reduction data generated in Section 3 is used to adjust the material properties in PDA, which are: 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 . These properties are determined so that the PDA prediction are as close as possible to the data.

In-situ values (F_{2t}^{is}, F_6^{is}) are adjusted until the PDA predictions match the initiation of modulus-reduction data (e.g., between 0.6 and 0.7 % applied strain in Fig. 8). The nominal in-situ strength values of the unidirectional lamina (F_{2t}, F_6) are used as initial guess for optimization. PDA's fracture energy (G_{mt}^c) is adjusted until the PDA prediction matches the modulus-reduction data (see Fig. 8). The critical energy release rate (ERR) for interlaminar damage of a similar material system is used as initial guess for optimization.

For IM7/MTM45-1 (Section 3.4), the $[0/90_4]_s$ laminate is analyzed first to adjust both F_{2t}^{is} and G_{mt}^c by minimizing the error between the PDA-predicted values and the modulus-reduction values obtained in Section 3.4. Next, any laminate with laminae $\pm\theta$ close to $\pm 45^\circ$ is useful to adjust F_6^{is} while G_{mt}^c and F_{2t}^{is} are kept unchanged. A $[0/\pm 55_4/0_{1/2}]_s$ laminate was used to adjust F_6^{is} . Results are shown in Table 4. The number of modulus-reduction data points has a little impact on the values obtained for the PDA properties as long as the stiffness degradation of the lamina is well represented (see Fig. 8).

Modulus versus the applied strain are shown in Fig. 8 and 9 for two IM7/MTM45-1 laminates. For laminate $[0/90_4]_s$ in Fig. 8, PDA results fit the experimental data nicely. For $[0/\pm 55_4/0_{1/2}]_s$ in Fig. 9, the error is larger. The discrepancy can be explained as follows. PDA does not include dissipation energy due to shear separately from transverse tension, but rather both are lumped into one term, i.e., G_{mt}^c . Thus PDA struggles when both modes, traction and shear, are combined in

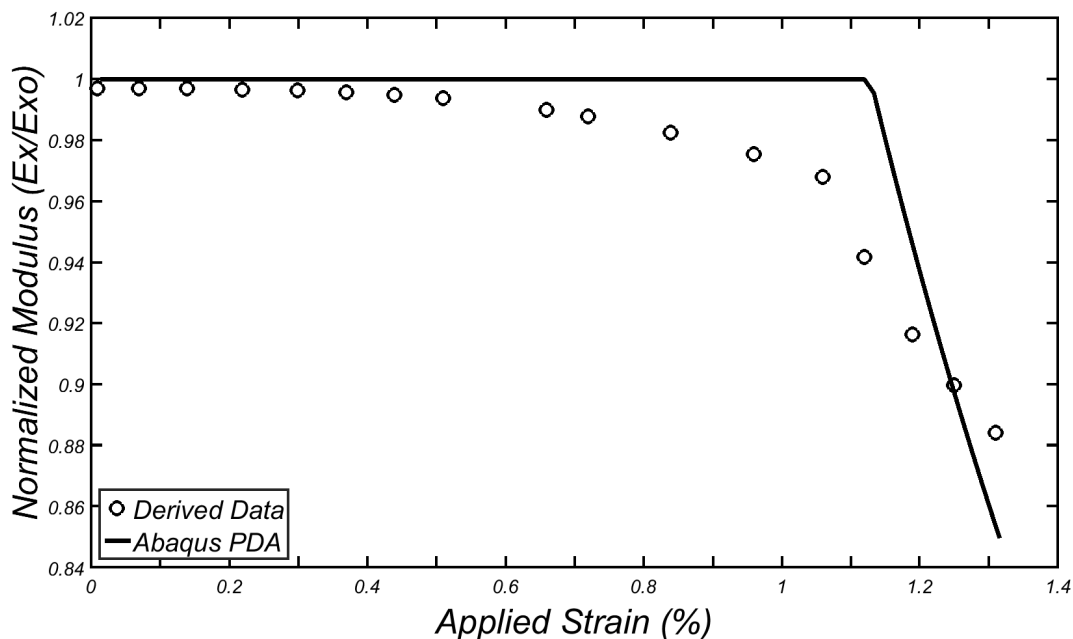


Figure 9: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode II for $[0/\pm 55_4/0_{1/2}]_s$ IM7/MTM45-1.

$[0/\pm 55_4/0_{1/2}]_s$. 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 (by the optimization algorithm) to start cracking at a higher strain than that shown by the data (open circles in Fig. 9). This is because the optimization function seeks to reduce the error (17) between the data points and the PDA prediction. Thus, increasing slightly the F_6^{is} value, reduces the error produced by the rest of modulus-reduction data points which belong to the *AC* side in Fig. 1. This would probably not happen if a shear dissipation energy parameter were computed separately from G_{mt}^c , but PDA does not have such parameter in its formulation.

Modulus versus applied strain are shown in Fig. 10-12 for IM6/Avimid (Section 3.1), T300/Fiberite 934 (Section 3.2), and AS4/Hercules 3501 (Section 3.3). For IM6/Avimid, laminate #2 was chosen to fit PDA results with modulus-reduction data as is illustrated in Fig. 10. The modulus reduction simulated using PDA matches nicely with experimental values and thus, obtaining a good G_{mt}^c adjusted to predict damage initiation and evolution.

For T300/Fiberite 934 and AS4/Hercules 3501, laminate #9 and #10 were chosen to fit with modulus-reduction data as shown in Fig. 11 and 12, respectively. For Figures 11 and 12, a significant deviation can be appreciated between experimental data and the modulus reduction using PDA. This difference is due to the PDA formulation (Figure 1). Once the in-situ strength F_{2t}^{is} is fixed to predict the onset of damage according with the Hashin criteria (Section 2.1), only one point in the triangle need to be adjusted to obtain the G_{mt}^c . This step is critical not only to calculate the final value but also it will set the modulus reduction rate for a specific material system. According to PDA formulation, crack density reaches saturation before DDM predictions, obtaining a damage variable equal to unit value. According to both PDA and DDM, no fiber breakage is predicted, thus reaching 1.08 % and 1.2 % of applied strain for #9 and #10 laminates, respectively. Note that strains higher than those shown in Figures 11 and 12 are not advisable in design, since other

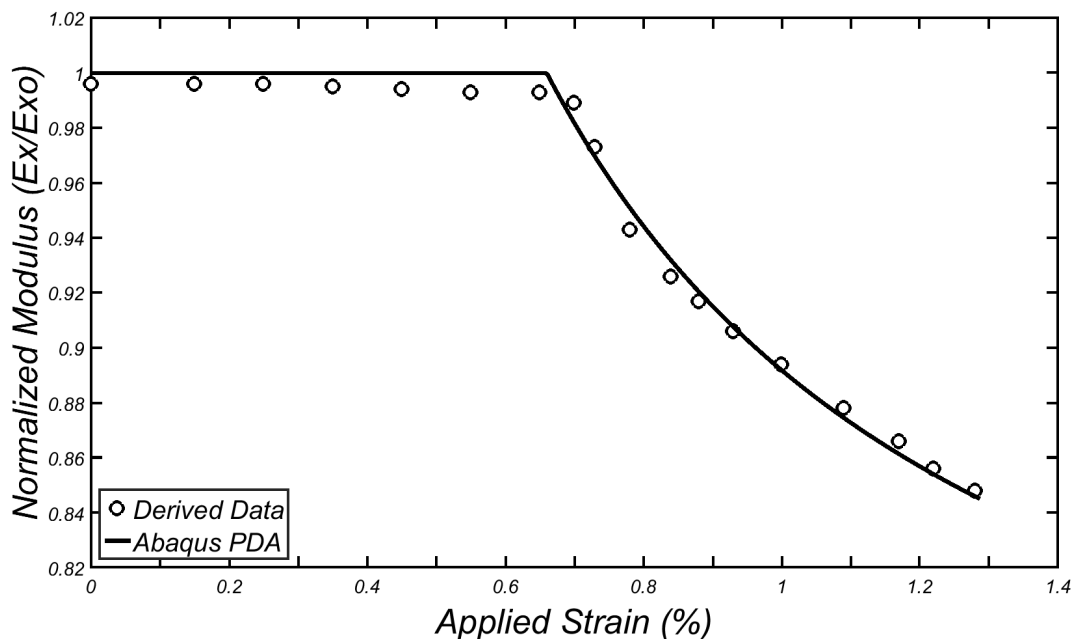


Figure 10: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_3]_s$ IM6/Avimid®K Polymer.

types of damage, such as fiber failure, may occur.

5 In-situ strength

The in-situ strength values can be calculated as shown in Section 4. Alternatively, they could be calculated with the methodology presented in [9, section 7.2.1]. Assuming a transition thickness $t_t = 0.8$ mm [20], for each of the four material systems the following calculations are performed and the results are shown in Table 5. The effective thickness is

$$t_e = \min(t_t, t_k) \quad (18)$$

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

$$F_{2t}^{is} = 1.12 F_{2t} \sqrt{\frac{2 t_t}{t_e}} \quad (19)$$

$$F_6^{is} = F_6 \sqrt{\frac{2 t_t}{t_e}} \quad (20)$$

where F_{2t} and F_6 are the unidirectional ply strength (Table 1). The comparison is shown in Table 5. Calculated values are accurate for only two of the four laminates. Therefore, adjusting them with the methodology proposed herein is a better choice when crack-density data is available.

6 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

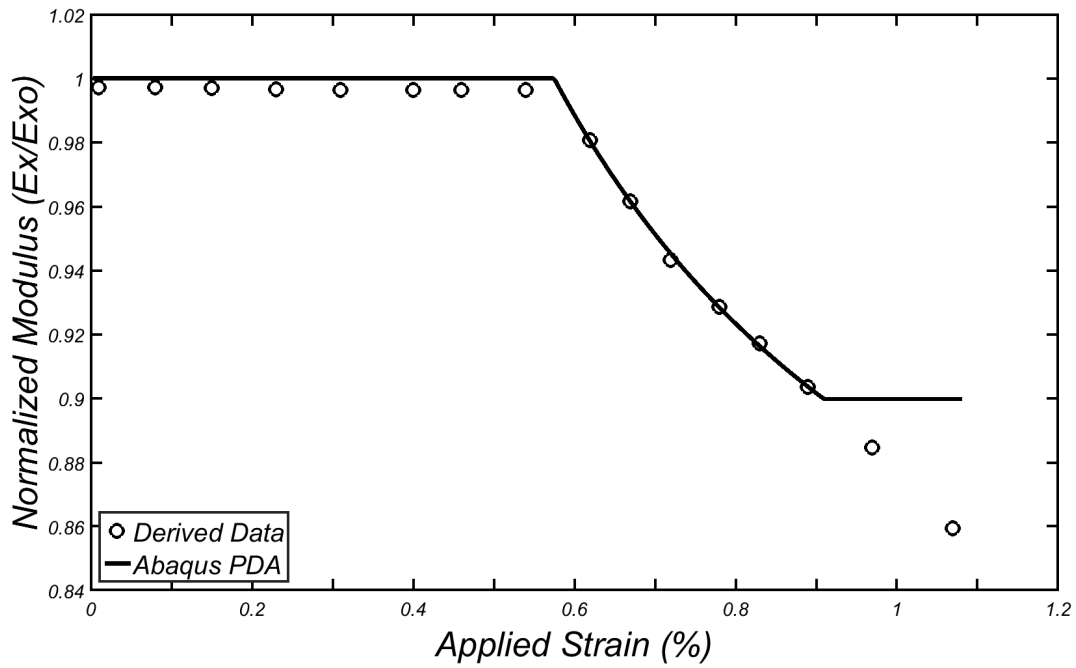


Figure 11: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0_2/90_4]_s$ T300/Fiberite 934.

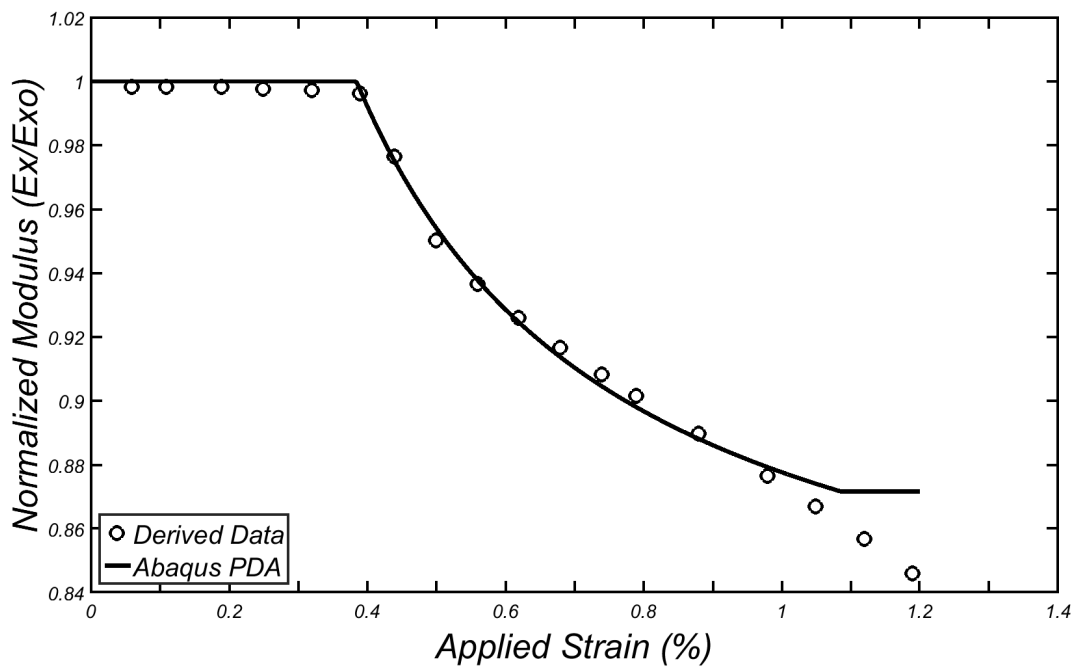


Figure 12: PDA model prediction vs. longitudinal modulus from derived data. G_{mt}^c is adjusted in mode I for $[0/90_2]_s$ AS4/Hercules 3501.

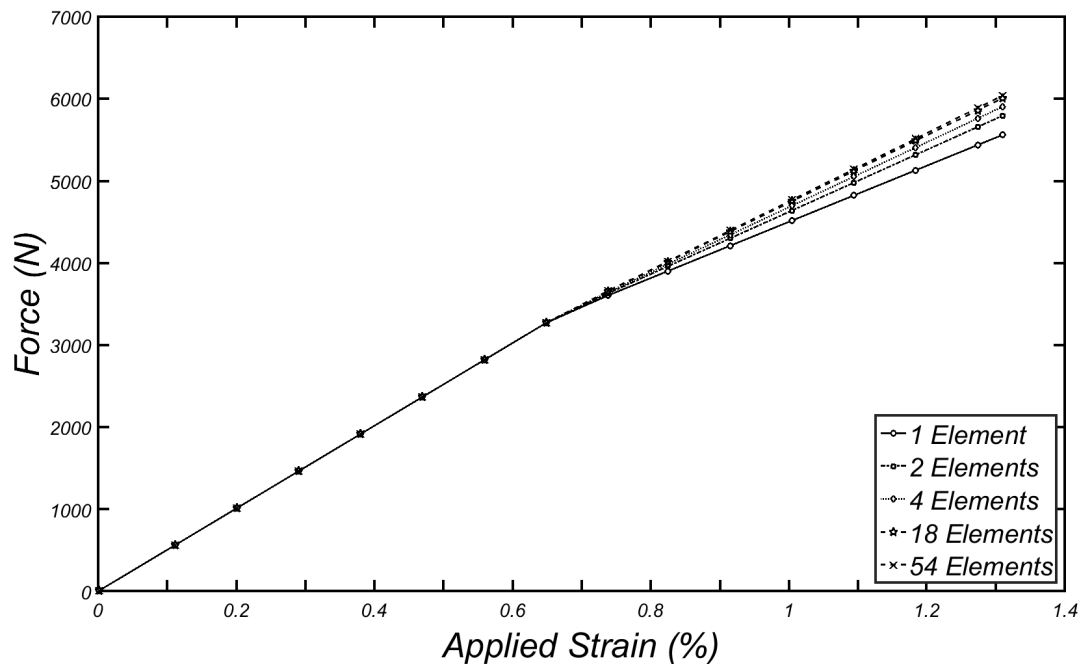


Figure 13: Force vs applied strain for laminate 12 as function of number of elements.

strain, the stress fields should converge independently of mesh refinement. However, the softening material behavior is mesh dependent, as it was mentioned in Section 2.2. Such mesh dependency can be observed in Fig. 13 and 14. The PDA softening force-displacements assumption (line ABC in Fig. 1) yields different results depending on the size of the elements as it can be seen in Figure 13. Consequently, the stiffness increases slightly with the number of elements, thus showing mesh dependency as shown in Fig. 14. Note that PDA material properties were adjusted using the data labeled “derived data” in Fig. 14, using a one-element discretization with one S8R element for the entire specimen. Despite the fact that the state of strain is uniform in the entire specimen, the normalized modulus seems to increase as the mesh is refined. Note that mesh refinement has no effect on the predicted strain because the strain is uniform for the whole specimen and the stress is uniform in each lamina. Any deviation in normalized modulus is due exclusively to mesh dependency of the PDA constitutive model.

The result of p-refinement, that is comparing S4R (linear interpolation) to S8R (quadratic interpolation) element types is shown in Figure 15. Using the properties adjusted with one S8R element into a simulation using one S4R element, sudden degradation of stiffness is observed with S4R. To solve this problem, one could use a finer mesh of S4R elements and/or find new values of the PDA material properties specifically adjusted for S4R. Such values as reported in Table 4 (using one S4R element). But this situation is very inconvenient for the user, who would be forced to change the values of the material properties depending on what type of element he wishes to use.

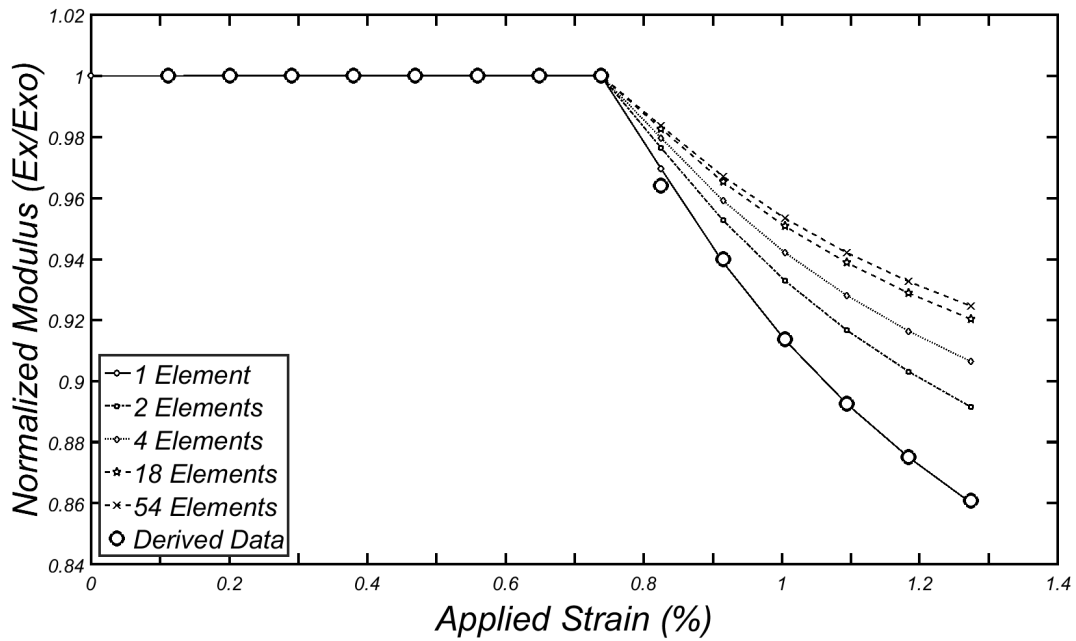


Figure 14: PDA model prediction vs. longitudinal modulus from $[0/90_4]_s$ laminate as function of number of elements.

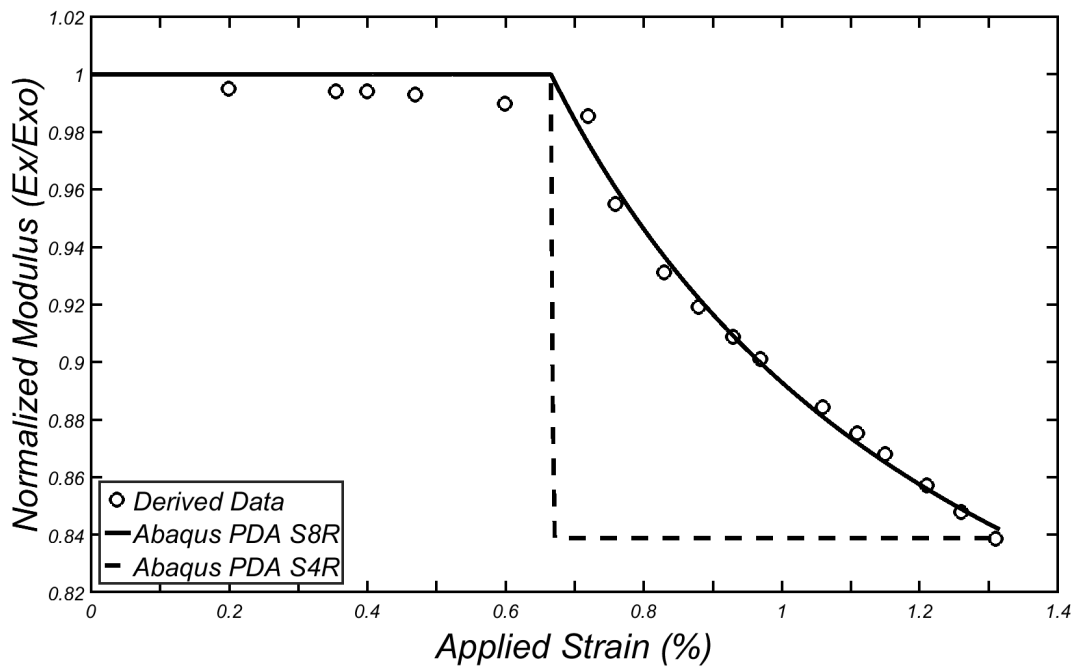


Figure 15: PDA model prediction vs longitudinal modulus from laminate $[0/90_4]_s$ as function of type of element.

7 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-reduction of the laminates. For Carbon/Polymer composites, a more elaborate method is needed because experimental modulus-reduction data for Carbon/Polymer has too low signal/noise ratio. To solve this problem, 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 modulus-reduction for those laminates. Finally, the calculated modulus-reduction is used to obtain the PDA properties by simply adjusting the PDA properties to obtain a good prediction of the modulus-reduction data. Using the proposed methodology, properties are calculated for four different material systems and subsequent predictions match observed behavior accurately. However, some shortcomings of PDA are observed. The results predicted by PDA are sensitive to mesh density and element-type used as was shown in Fig. 13, 14, and 15. Also, PDA properties vary depending on type of element used (Table 4). These are evidence of mesh sensitivity, also called lack of objectivity. Furthermore, lack of shear dissipation energy in PDA makes it difficult for PDA to accurately predict the response of laminates where damage is due to mainly in-plane shear. This is notable for laminates with cracking laminas close to 45 and for laminates without laminas at 90 (Fig. 9). These shortcomings cannot be ameliorated without modifying the PDA formulation and associated source code, which is not accessible in commercial codes such as Abaqus.

Appendix. Software Installation

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 as follows:

- Determine the Python (\gg import sys) and *Numpy* version (\gg import *version*) using the windows command for the installed version of Abaqus.
- Install the correct Python version determined previously. Onwards, any necessary library to be used by Abaqus except *Numpy* library must be first installed in the Python folder. Note that the *Numpy* library is already installed in Abaqus by default, so the *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), which by default is installed in the Python folder. 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). The *Scipy* library contains the optimization and advanced mathematical functions.
- Once the *Scipy* library is installed in Python folder, it must be copied/moved to the Abaqus library folder. After that, it is possible to import any function from *Scipy* library. This procedure can be followed for any other type of library if needed.

The rest of the problem is limited to write three Abaqus scripts (*available as supplemental material for this paper*) and run them directly from CAE. Each script plays an important role and

must be located in the same folder as the Abaqus work directory. Then, each script can be called through the *import* command. The functionality of each script is 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 geometry, 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 equation (16) to match laminate stiffness degradation data is defined in the second script. Note that the experimental data are written inside the code to reduce execution time, and the class model from *Script1* is imported. Note that the error function is named *Iteration* in *Script2*.
- Finally in the third script, the error function is imported from *Script2* and the optimization problem is solved using the *fmin* function from *Scipy*. Then, options and tolerances are adjusted, and the results (function value, number of iterations and function evaluations) are printed on the Abaqus window. Only *Script3* must be run from *Abaqus* to start the optimization.

References

- [1] E. Barbero, F. Cosso, R. Roman, T. Weadon, Determination of material parameters for Abaqus progressive damage analysis of E-glass epoxy laminates, *Composites Part B: Engineering* 46 (2013) 211–220.
- [2] P. P. Camanho, C. G. Dávila, Mixed-mode decohesion finite elements for the simulation of delamination in composite materials, *NASA/TM* (2002) 1–37.
- [3] A. Matzenmiller, J. Lubliner, R. Taylor, A constitutive model for anisotropic damage in fiber-composites, *Mechanics of materials* 20 (2) (1995) 125–152.
- [4] Abaqus online Documentation, <http://129.97.46.200:2080/v6.14/books/hhp/default.htm>.
- [5] S. Liu, J. A. Nairn, The formation and propagation of matrix microcracks in cross-ply laminates during static loading, *Journal of reinforced plastics and composites* 11 (2) (1992) 158–178.
- [6] J. Varna, R. Joffe, N. Akshantala, R. Talreja, Damage in composite laminates with off-axis plies, *Composites Science and Technology* 59 (14) (1999) 2139–2147.
- [7] J. Varna, R. Joffe, R. Talreja, A synergistic damage-mechanics analysis of transverse cracking in $[\pm\theta/90]_4$ s laminates, *Composites Science and Technology* 61 (5) (2001) 657–665.
- [8] A. M. Abad Blazquez, M. Herraes Matesanz, Acoustic emission characterization of intralaminar damage in composite laminates, Master's thesis, Carlos III University, Spain (2013).
- [9] E. J. Barbero, *Introduction to composite materials design*, 3rd Edition, CRC press, 2017.
- [10] E. J. Barbero, J. C. Barbero, C. N. Ugena, Analytical solution for plane stress/strain deformation of laminates with matrix cracks, *Composite Structures* 132 (2015) 621–632.
- [11] J. A. Nairn, The strain energy release rate of composite microcracking: a variational approach, *Journal of Composite Materials* 23 (11) (1989) 1106–1129.
- [12] J. A. Nairn, S. Hu, J. S. Bark, A critical evaluation of theories for predicting microcracking in composite laminates, *Journal of Materials Science* 28 (18) (1993) 5099–5111.
- [13] S.-R. Kim, J. A. Nairn, Fracture mechanics analysis of coating/substrate systems: Part ii: Experiments in bending, *Engineering fracture mechanics* 65 (5) (2000) 595–607.
- [14] C. V. Singh, R. Talreja, Evolution of ply cracks in multidirectional composite laminates, *International Journal of Solids and Structures* 47 (10) (2010) 1338–1349.
- [15] E. J. Barbero, D. H. Cortes, A mechanistic model for transverse damage initiation, evolution, and stiffness reduction in laminated composites, *Composites Part B: Engineering* 41 (2) (2010) 124–132.
- [16] Z. Hashin, A. Rotem, A fatigue failure criterion for fiber reinforced materials, *Journal of composite materials* 7 (4) (1973) 448–464.
- [17] E. J. Barbero, Prediction of compression strength of unidirectional polymer matrix composites, *Journal of composite Materials* 32 (5) (1998) 483–502.

- [18] E. J. Barbero, *Finite element analysis of composite materials using Abaqus™*, CRC press, 2013.
- [19] E. J. Barbero, F. Cosso, Determination of material parameters for discrete damage mechanics analysis of carbon-epoxy laminates, *Composites Part B: Engineering* 56 (2014) 638–646.
- [20] A. Wang, Fracture mechanics of sublaminar cracks in composite materials, *Journal of Composites, Technology and Research* 6 (2) (1984) 45–62.
- [21] 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* 93 (2) (2011) 1021–1030.

Table 1: Unidirectional ply properties for laminates IM6/Avimid K, T300/Fiberite 934, AS4/Hercules 3501-6 and IM7/MTM45-1 [5, 8, 18, 21].

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

Table 2: Laminates considered in this study.

Laminate	Stacking Sequence	Material
1	$[0/90_2]_s$	IM6/Avimid K [5]
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 [5]
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 [5]
11	$[0_2/90_2]_s$	
12	$[0/90_4]_s$	IM7/MTM45-1 [8]
13	$[\pm 25/90_5]_s$	
14	$[0/\pm 55_4/0_{1/2}]_s$	
15	$[0/\pm 70_4/0_{1/2}]_s$	

Table 3: Critical energy release rates for Discrete Damage Mechanics (DDM) model [19].

DDM Properties	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

Table 4: In-situ strengths and PDA fracture energy G_{mt}^c .

Property	Units	IM6/Avimid K		T300/Fiberite 934		AS4/Hercules 3501		IM7/MTM45	
		S8R	S4R	S8R	S4R	S8R	S4R	S8R	S4R
Carbon/Epoxy									
F_{2t}^{is}	[MPa]	64.5926	64.5896	41.2506	41.3186	35.4957	35.4320	50.9081	50.9103
F_6^{is}	[MPa]	-	-	-	-	-	-	128.4921	128.8867
G_{mt}^c	[KJ/m ²]	5.1528	10.3045	3.9838	8.1693	4.8687	9.7574	12.2171	21.1091

Table 5: 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

Supplemental materials

Script 1

```

### Script1 ###

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 *

# IT IS DEFINED A NEW CLASS MODEL: SET UP, RUN JOB AND GET THE RESULTS

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, ('[#85 ]', )),
), )

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

# THE TOTAL FORCE IS RETURNED

```



```

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.
# 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 HyE9082Af')
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 HyE9082Af'].HashinDamageInitiation(
table=((1020.0, 620.0, F2strenght, 140.0, 48.5725, 70.0), ))
mdb.models['Model-1'].materials['glass/epoxy Fiberite HyE9082Af']
].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 HyE9082Af', plyName='K1'), SectionLayer(
thickness=0.576, orientAngle=90.0,
material='glass/epoxy Fiberite HyE9082Af', 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=135.0)
# 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

```
### Script2 ###

from Script1 import Model
import math
# [Gmc, F2t, E(MPa), Area(mm^2), l_o(mm), Tol(mm)]    VARIABLES AND
# ARGUMENTS IN ORDER TO OPTIMIZE
def Iteration(x, *args):
    Gmenergy = x[0]
    F2strenght = x[1]
    E = args[0]
    Area = args[1]
    l_o = args[2]

    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
}
# FUNCTION ERROR. INITIAL VALUE
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

```

### Script3 ###

# We must run this script to optimize the variables Gmenergy and F2strenght

import numpy as np

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

from scipy.optimize import fmin

from Script2 import Iteration

#Gmenergy = 3 # Toughness fracture (initial value)
#F2strenght = 31 # Transverse strength in-situ (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)
```