TU DELFT - FACULTY OF AEROSPACE ENGINEERING

Semi-analytical approach to multiple delaminations in impacted composites

Jakob Kurpierz, 1397249

11/30/2012

Table of Contents

1 Introduction

The amount of continuing publications on composite mechanics shows that research into this particular field of engineering is still very popular. This is due to mainly two aspects: firstly, the complexity of the mechanics of composite materials provides many interesting and challenging research areas and secondly the increase in range and frequency of applications of composite materials to solve engineering problems demands fresh answers. In this thesis an attempt is made to shed some light on a specific aspect of composite materials which is inherent to the synthesis of these types of materials, namely delaminations.

A more detailed motivation for the research is given below after which the research goal and objectives are outlined. An overview of the entire thesis outline will conclude this chapter.

1.1 Motivation for the research

The increased use of composite materials in many fields of engineering, but particularly in the field of aeronautical engineering, has been the main driver for research into composites and their mechanics. In civil aviation this has recently lead to Boeing's Dreamliner, which structural weight consists of more than 50% of composite materials and the Airbus A380 (see Figure 1.1), where large parts of the wings an upper fuselage (about 20% of structural weight) are manufactured from composite materials [4].

While the use of composite materials generally promises to be more efficient than current metal

Figure 1.1 - Airbus' A380 (left) and Boeing's Dreamliner (right) (Courtesy of Airbus and Boeing)

designs there are some pitfalls related to the damage mechanisms inherent to composite materials. A composite laminate is made up of several plies, in which the fibres are oriented at different ply angles with respect to a global coordinate frame to suit the strength and/or stiffness requirements of the product. Those plies are either pre-impregnated or dry, which means that, in the latter case, the second constituent of a composite material, the matrix, still needs to be added. While the fibres provide load carrying capabilities the matrix makes sure that the fibres are held in place. Owing to the nature of the material there are several failure modes related to the different constituents of a composite material [5]:

- matrix cracking, which occurs at intraply level and often is initiated at defects or fibre matrix interfaces
- fibre failure, which occurs when the strength in the fibres is surpassed

- delamination, which indicates the separation of individual plies or lamina due to matrix failure at the interply level

The focus of the current thesis lies on the last item, delamination. It is the most severe form of internal damage in laminates when flexural applications are concerned, since it effectively separates the entire structure into two or more substructures resulting in decreased flexural stiffness. The title of the thesis also gives away the second important aspect, which is composites under impact. While impact generally can mean anything from a screw dropped from a few meters height to the ballistic impact of projectiles, the focus here lies on the former. During manufacturing and assembly it happens all too often that tools are dropped onto parts of the structure which can result in performance-decreasing damage that is invisible to the naked eye and is not considered important as the dropped tool is picked up. This is another important problem with composites. While the impact might not have left a visible mark on the impact surface, the structure underneath could have potentially undergone severe internal damage in the form of matrix cracks and delaminations. Due to this rather complex, and not entirely understood damage behaviour, composite materials are often penalized during design stages resulting in marginally better than or equal performances to their isotropic counterpart. It is therefore considered essential to develop a thorough understanding of the formation and interaction of such internal damage states using an analytical approach to the problem to enhance the use of composite materials during design and also during the service life of a part in connection with maintenance, etc.

To date (2012) composite parts susceptible to impact damage are evaluated using either experimental tests or finite element analyses. While these two approaches will certainly yield highly accurate results in terms of the damage state they are very expensive to carry out. Both need a lot of time to set up and many other resources such as testing personnel, materials and equipment and considerable computing power. Obviously, an analytical approach will not be able to replace the above mentioned practices. Rather it should be used to help engineers make preliminary decisions based on which best candidate designs are selected for further, more accurate, evaluation using either experiments or finite element analyses can be carried out.

Ultimately, the results of this research should lead to more impact damage resistant/tolerant composite design guidelines, which combined with existent knowledge and guidelines [4] can be used in the preliminary design stages of composite parts.

Finally, it should be mentioned at this point that this thesis originated from a cooperation with PhD candidate Mohamad Talagani MSc and his thesis entitled “Damage tolerance of advanced composite structures”.

1.2 Research tasks and objective

The thesis is built around the research objective formulated as follows:

To develop a (semi-)analytic model for multiple delaminations in an impacted composite plate by obtaining an accurate representation of the stress field near the impact zone, using failure criteria in combination with fracture mechanics concepts to determine conditions for delamination initiation and growth, and to investigate the interactions between delamination zones.

Several research tasks can be extracted from this objective, which form milestones in the execution of the project:

- obtain force and indentation history for quasi-static variants of impact events

- obtain accurate stress values from the contact due to impact
- determine first delamination site and establish growth criteria
- determine updated stress field under presence of cracks
- determine subsequent delamination sites and investigate their interaction with each other

Each of these tasks will be explained in more detail in the respective subsections of the main chapter of this report dealing with the analytical model. While each and every one of these tasks is important the synthesis of these separate aspects into one model lies at the core of this thesis. The synthesis is based on a modular and sequential approach, with the result from one analysis flowing directly into the next the step. This will be discussed in detail in chapter 3.

1.3 Outline of the report

The remainder of the report follows a natural sequence. A brief summary of the research done in the various fields of interest in this research is given in chapter 2. The main part of the work is described in chapter 3, which consists of a description of the analytical model. The modularity of the approach is explained and each of the research tasks are treated in detail to obtain the analytical model. The application of this model to a few test cases and its potential use in an optimization scheme with respect to damage resistance/tolerance is discussed in chapter 4. Finally, conclusions are drawn and recommendations are given in chapter 6.

2 Literature Review

The literature related to the topics involved in this thesis is presented in this chapter. This thesis deals with the impact behaviour of composite structures. The impact considered is assumed to be quasi-static and therefore is able to represent a tool dropped, an impact event during manufacturing or maintenance processes, for example. Impact events such as tool drops can lead to internal damage creation within in the laminate in the form of multiple delaminations which might not be detectable with the naked eye. Such delaminations can be detrimental to the structure in case of compressive or bending applications, since they reduce the bending stiffness of the impacted structure.

This chapter is split into four sections; one, the literature on existing solutions to the impact problem is discussed, two, the literature on obtaining accurate stress states for undamaged and damaged stacks is covered, three, the literature concerning the damage and fracture mechanics of composites is covered and in section four the literature on existing approaches to predicting damage in impacted composite structures is discussed. Each of the first three sub-objectives has been covered extensively in the literature and a good overview up until the turn of the millennium is provided in Abrate [6], which contains lot of detail on the specific problems pertaining to obtaining the force history during an impact event. The remaining two sub-problems are treated less thoroughly. Abrate relies entirely on plate theories to predict the transverse stresses in impacted composite plates and discusses damage prediction only qualitatively.

2.1 Impact problem

A variety of solutions procedures exist to solve the problem of a force acting perpendicular to the surface of the structure, which is how the tool drop can be simulated.

In the context of the first order shear deformation plate theory (FSDT) Dobyns [7] investigated the behaviour of simply supported orthotropic plates under static and dynamic loads. Due to the nature of the problem the flexural displacements are only given in terms of infinite series and the solution of the impact problem was obtained by solving a non-linear integral equation. The Hertzian contact law (see chapter 3 for more detail) was used when modelling the impact problem. The impact load could take the shape of a concentrated force, be spread over a small area, etc. In case of a load spread over a small area it could be shown that the peak transverse shear forces acting on the plate occurred at the boundary of the load patch. While the solution to the problem was successfully obtained it lacks the simplicity of being implementable in straightforward manner.

In order to account for the well-known Hertzian contact formulation during impact, several spring-mass models have been set up and investigated as to reproduce the impact event as accurately as possible. A two degree of freedom (TDOF) spring mass model was proposed by Shivakumar, Elber [8] for the analysis of the impact force and duration during low velocity impact. Impactor and an equivalent plate mass act as the two degrees of freedom whereas springs represent the deformation between impactor and plate and the plate itself, respectively. Bending, transverse shear and membrane deformation of the plate can be modelled by solving the two resulting non-linear differential equations numerically. For large impactor masses relative to the equivalent plate mass, the model simplifies to a single degree of freedom model (SDOF), which still needs to be solved numerically due to the presence of the non-linear contact formulation. Besides this improved spring-mass model they also suggested a rather simple energy-balance system, which provides a quick yet accurate means of predicting the peak force during impact. If all contributions

are to be taken into account, then the resulting equation can be solved by a Newton-Raphson approach.

A simplified version of the analysis done by Shivakumar has been carried out by Pang and his colleagues [9]. They took an adapted spring-mass model representing thin plate small deflection theory. By assuming a harmonic expression for the force acting on the plate they ended up with a single non-linear equation which still needed to be solved numerically. Nevertheless, it greatly enhanced the complex analysis from Shivakumar, Elber [8].

From a one-parameter differential equation Olsson [10] managed to derive the impact response of a specially orthotropic plate for the first impact phase. This includes the part of the impact event where the highest contact forces occur before the interaction of flexural waves reflected from the boundaries. Again, Hertzian contact formulation is used and the result of the dimensionless differential equation can be used to express contact force, deflection and pressure. The only parameter influencing the ordinary differential equation is the so-called inelasticity parameter lambda, which depends on the impact and plate configuration. In his paper he explains the fundamental difference between boundary- and wave-controlled impact response. In the case of the boundary-controlled impact the entire plate is deformed during impact and depending on the ratio of contact duration to lowest vibration mode the response can be considered quasi-static. During wave-controlled impact the deformations are localized around the impact event.

Following his paper published in 1992, Olsson [2] sets out to define a mass criterion for the wave-controlled impact response of composite plates. Rather than using the impactor velocity to define the type of impact response as done by Shivakumar, Elber [8] he states that the impactor-plate mass ratio is a better suited classification criterion. This relates to the fact that different masses impacting with the same velocity inflict completely different damage states. The appropriate analytical model needs to be chosen for each of the three different cases, small-, intermediate- and large-mass. In the case of a small-mass Olsson succeeded in finding the limiting impactor-plate mass ratio. He also discusses the limits of large-mass impacts and compares these to those found by Swanson in his paper [11]. Swanson uses the fact that, for quasi-static cases, the knowledge of the displacement field is sufficient to determine the lumped mass of the impacted structure and that for a ratio of projectile mass to lumped mass of larger than 10, the response is considered to be quasi-static.

2.2 Stress determination

Accurate knowledge of the complete stress state at any given moment throughout the impact event is paramount for the successful prediction of delamination sites. Naturally, significant attention has been paid to this particular topic and a variety of approaches have been put forward, of which a selection will be presented in the following.

The difficulty regarding delaminations and any preceding failure modes during the impact event such as matrix and shear cracking lies in the fact that they are all triggered by interlaminar shear and normal stresses. Most solutions regarding the mechanics of composites are deduced from the plane stress or strain reduced equilibrium equation. The much needed transverse stress components are then recovered from integrating the full set of three dimensional equilibrium equations [12]. The transverse shear stress components then follow a parabolic, or in the case of composites, a piecewise parabolic distribution with the maximum located at midplane. During the impact event, however, large contact stresses occur locally, which need to be accounted for. Building upon Boussinesq's solution to the problem of a concentrated force acting on a semi-

infinite body, Love [13] investigated semi-infinite bodies under pressure acting on parts of the boundary. He used the Hertzian contact formulation to express the entire stress state in closed form for an isotropic semi-infinite body resulting from contact with a hemispherical impactor. The equivalent version to the problem solved by Boussinesq for transversely isotropic material is provided by Lekhnitskii in his work on anisotropic elasticity theory [14]. This provides the basis for Dahan and Zarka [15] who using the Hertzian contact formulation successfully delivered the transversely isotropic counterpart for Love´s solution.

Using the available literature a simplified approach to the impact problem has been set up by Olsson and Nilsson [16]. They combine the solution of a simply-supported circular Kirchhoff plate with the contact stresses in a transversely isotropic semi-infinite body according to Dahan and Zarka both due to the Hertzian contact load. Naturally, the solution to the second part is only approximate; however, it was shown that the agreement between the predictions and conducted FE-analysis was good.

Common to all of these approaches is the fact that a semi-infinite body has been considered, i.e. that interlaminar stresses were specified to be zero at infinity allowing for analytical solutions to the integrals involved. Contrary to this, Cairns discussed the problem of a finite thickness plate in his PhD thesis [17]. He solved the problem by expressing the stress function using a Fourier-Bessel series and satisfying the boundary conditions at both the impact and back surface. While achieving the desired results, the solution can only be obtained numerically and its accuracy depends on the number of terms used in the series expansion. A different approach based on Boussinesq's equations has been investigated by Talagani [18]. Using these in combination with the Hankel integral transform leads to an integral expression for the stress components similar to those presented by Dahan and Zarka [15]. Due to the fact that the plate has a finite thickness these can only be solved numerically. The results are qualitatively similar to those obtained by Cairns and have been satisfyingly compared to finite elements (FE) results. Again, however, the accuracy of the results depends on the numerical solution method used. Both methods mentioned here constitute semi-analytical approaches valid for transversely isotropic material.

While all of the approaches listed above have been approved for structures, which have not yet suffered any internal damage, it cannot be assumed that they are equally valid for bodies featuring delaminations or similar faults. Solutions for the case of a delaminated structure subject to a load acting perpendicular to its surface have not been found in the literature on elasticity theory or related topics, which therefore lead the literature search to be extended into the realms of fracture mechanics.

2.3 Damage/Fracture mechanics

On an interlaminar level damage normally first occurs in the form of transverse matrix cracks., which arise when the introduced transverse shear stress surpasses the shear strength of the matrix. Eventually these cracks reach a ply interface and at their tips high stress intensities are dominant. These stress intensities initiate delaminations. This means once the interlaminar stresses are determined to a satisfying degree of accuracy, a failure criterion can be applied to determine where exactly the first and any subsequent delaminations are initiated. Once that is known linear elastic fracture mechanics (LEFM) approaches can be used to predict growth of the delaminations.

A comprehensive review of failure criteria has been published by Orifici, Herszberg [5]. They cover individual laminae constituent failure modes criteria and delamination initiation and growth criteria. The problem with many of the failure criteria is that they do not take into account the sign of the transverse normal stress with regard to the initiation of delamination. One of the first to

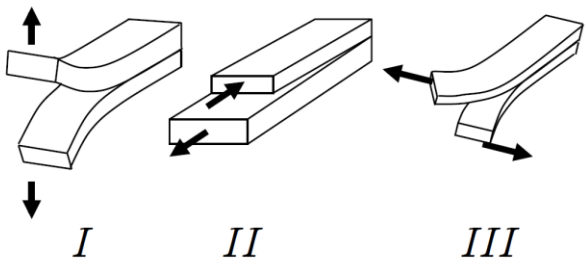


Figure 2.1 - Basic modes of delamination growth. Opening mode (I), sliding mode (II), tearing mode (III)

publish delamination initiation criteria was Hashin [19] after which a range of criteria followed. One such failure criterion, which was used successfully to predict the onset of delamination due to in-plane loads, was derived by Brewer and Lagace [20]. They used average transverse stress values in combination with the respective transverse strengths of the material in a quadratic stress criterion. Comparing it to the energy release rate approach it was found that their stress criterion was able to predict the onset of delamination more accurately for a variety of laminate layups. However, the averaging length and material strength parameters would have to be determined first. Taking into account the sign of the transverse normal stress Hou, Petrinic [21] devised a criterion which allots a beneficial influence of compressive normal stresses to the formation of delaminations and would only allow one to form if the shear stresses are sufficiently large.

Once a crack in from of a delamination has been found within the laminate the concepts of LEFM can be applied to study its evolution. Numerous articles and books have been published on this topic of which a selection will be presented here. A recent and comprehensive textbook is that of Sun and Jin [22]. There are basically two distinct approaches to fracture mechanics: the local approach involving the near crack tip stress fields and the global energy balance approach. The important factor in the first approach is the stress intensity factor (SIF) while for the second approach the energy release rate (ERR) plays a major role. Both can be used to study the growth of a delamination as they are related to each other and therefore often appear in the same publications. Equivalent to the energy release rate is the so-called J-integral set up by Rice [23]. This integral is path independent, which means that an arbitrary path enclosing the crack can be taken to yield the same result, offering great opportunities to the researcher.

Expressions SIFs can be obtained for a wide range of applications from handbooks [24]. However, nearly all of them are for isotropic homogeneous cases and the particular case of a structure under a perpendicular concentrated load with an embedded crack does not seem to be covered. Fortunately, there have been some interesting contributions on stress intensity factors at interfaces between dissimilar media. The complicating factor when studying interfacial stress intensity factors lies in the fact that the different delamination modes (see Figure 2.1) all occur simultaneously. This leads to stress oscillations close to the crack tip [22]. Sun and Jih [25] who also struggled with these phenomena, have studied the energy release rate for interfacial cracks between two different material layers, both analytically and numerically. They managed to express the energy release rate for the individual modes in terms of the complex interface stress intensity factor. It was found that due to the inherent oscillation the individual terms did not correspond well with FE results. However, the sum of the individual modes, which is the total energy release rate for the crack, was very well defined. While Sun and Jih's paper was restricted to isotropic bi-materials Hwu and Hu [26] extended their idea to general anisotropic material resulting in the same oscillatory behaviour in the definition of the individual energy release rates. Both papers depend on obtaining the solutions for the stress intensity factors from the asymptotic stress field around the crack. Suo and Hutchinson [1] have defined these interface stress intensity factors for a semi-infinite crack between two infinite isotropic layers under general edge loading. It is possible to define the complex stress intensity factor for the interface by expressing the loading at the crack tip in terms of the stress resultants. The only loading considered was bending and in-plane normal loads.

Extending these ideas to allow for shear deformation has been done Wang and Qiao [27]. They use Reissner-Mindlin plate theory which allowed them to express the complex stress intensity at an interface between two orthotropic elastic layers in closed form. A further extension with respect to the number of contributing terms is provided by Andrews [28]. He introduces root rotations next to the shear deformations in the computation of the stress intensity factor due to an equivalent crack-tip loading system, similar to the one used by Suo and Hutchinson [1]. Together with the expressions he presents guidelines for the validity of these expressions in terms of minimal distances ahead and behind the crack tip. It has to be noted, though, that his expressions have not been derived for interface cracks and that they rely on numerically computed root rotation compliances.

Adopting the global energy balance approach Suemasu and Majima [29] have investigated the severity of multiple delaminations and their growth in a circular composite plate due to a transverse load. Assuming disc shaped delaminations they found closed form expressions for the critical force at which the delaminations would start to propagate. The energy release rate was found to be independent of the delamination size. They also discussed the event of a single delamination being either longer or shorter than the remainder of the delaminations and found that there is a tendency for the single delamination to adjust to the rest of the delamination pack. In their derivation they only considered equally spaced delaminations.

Interesting research in the particular field of delamination onset and growth under impact loading has been carried out by Olsson. In one of his papers [30] he investigates the delamination growth of composite plates under quasi-static impact loading for multiple delaminations and is able to assign bending, shear and membrane contributions. In a similar fashion [31] he has studied small-mass impact behaviour of composite plates. He combines his results with the expression for static delamination threshold force in order to arrive at the delamination threshold velocity and compares them successfully with results from the literature. He then continues to find a relationship for the delamination threshold load for the dynamic impact [32] and demonstrates that the inclusion of dynamic terms leads to an increase of that load. He was also able to derive the delamination threshold velocity through an iterative process. He concludes that for material systems with low transverse shear stiffness this velocity is under-predicted. It has to be emphasized that in the above mentioned references, Olsson always assumes that the delamination will start at the mid-plane, several delaminations will always be spread evenly throughout the stack and the material in which the delaminations occur is homogenized.

Since multiple delaminations will be considered in this work another aspect which has not been covered extensively in the literature had to be researched, the interaction of two or more delaminations within a laminate. Zheng and Sun [33] proposed a triple plate finite element model and investigate the interaction between two delaminations in a three point bending specimen as well as in a circular plate subjected to a central load. In the case of the circular plate they have shown that the interaction between the two delaminations depends strongly on the relative position of the delaminations to concurrent delaminations. A major effort has been made by Andrews who has investigated the elastic interaction between multiple delaminations [34]. He did this for plates undergoing cylindrical bending using, in total, three different contact formulations between the separated layers. Interaction effects such as shielding or amplification, already found by Zheng and Sun, were encountered, and for a system with two delaminations these can be visualized in graphical form. The validity of these findings has been demonstrated and it was founds that results are well behaved if the tips of the delaminations kept at a certain minimal distance apart.

In contrast to the stress-based investigation of damage detailed above, Liu [35] stated that delaminations occur because of bending stiffness mismatches at an interface of two differently orientated laminae. He bases his assumption on the fact that the impact problem can be viewed as mainly bending stress influenced. In his hypothesis he derives a bending stiffness mismatching coefficient which is proportional to the delamination area. Furthermore, Liu has studied the dependence of material properties, layup sequence and laminate thickness on the delamination area and, based on these findings, has demonstrated how to arrive at a damage resistant laminate.

2.4 Damage prediction due to impact

In trying to combine the separate aspects of finding the failure sites in laminates with predicting delamination interaction and growth several authors have set out to build damage prediction models for impacted composites. These are either based on complex FE analyses or extensive test programmes, both of which approaches are extremely time and resources consuming.

Choi and Chang [36] use the FE approach to predict internal damage, that analytical models are not capable of predicting. They use matrix and shear cracks to initiate delaminations and distinguish between two distinct propagation cases based on the preceding failure mode. They propose a stress-based delamination growth criterion featuring a parameter that has to be obtained experimentally and is assumed to be only material system dependent. While giving reasonably successful correlation between test data and FE results their model was not able to incorporate interaction between multiple delaminations or any form of material degradation.

In the context of an internationally conducted research project it was found that there is no strategy that is able to be used to predict the compressive residual strength due to the incident impact energy. It became necessary to find methods that can be used to predict the internal damage first. In their paper Davies and Zhang [37] present several coupon tests and predict both damage size and threshold load using gross simplifications of attributing isotropic behaviour to the composite structures investigated. They found that a degradation formulation for their numerical tool was necessary for accurate predictions. Such degradation models in combination with a cohesive zone model can be found in almost any publication on the numerical prediction of delaminations and other types of failure modes (Davies, Hitchings [38], Koloor, Abdul-Latif [39], Lopes [40]). Lopes' work highlights both the complexity involved in setting up such an FE model and the time needed for just one impact simulation and even so the correlation of the results with experimental results can still be improved upon.

An interesting concept observed by several authors and systematically researched by Schoeppner and Abrate [3] is that of the delamination threshold load (DTL). They use a broad set of impact test data, which contained three different material systems and a variety of layups of different laminate thicknesses. Based on this extensive experimental evidence they were able to relate the force at which significant internal damage will occur to the thickness of a laminate for each of the material systems considered. Therefore, this concept can be used to serve as first indicator of a laminate's damage resistance to low-velocity impact. If the maximum contact load from a quasi-static approach is below this threshold load it can be assumed that no significant damage will occur. However, no statement has been made regarding the validity of this approach for other structures than those investigated and reported in [3], which conform to the ASTM standard for impact tests [41].

The aim of the current work, reported here, was to complementing what has been done in this field with an emphasis on providing a fast and accurate solution in terms of damage prediction, which does not require the usage of complex and expensive numerical tools.

3 Analytical model

The development of an analytical model to determine the damage state in an impacted composite structure was the main effort of this thesis and is the focus of this chapter.

3.1 Methodology outline

The emphasis at the outset of this project was to create an analytical model, which would be capable of predicting initiation and growth of multiple delaminations in impacted composite structures. While this comprises a range of engineering disciplines the aim was always to keep the model simple, and therefore fast, yet it should be able to capture the actual development of

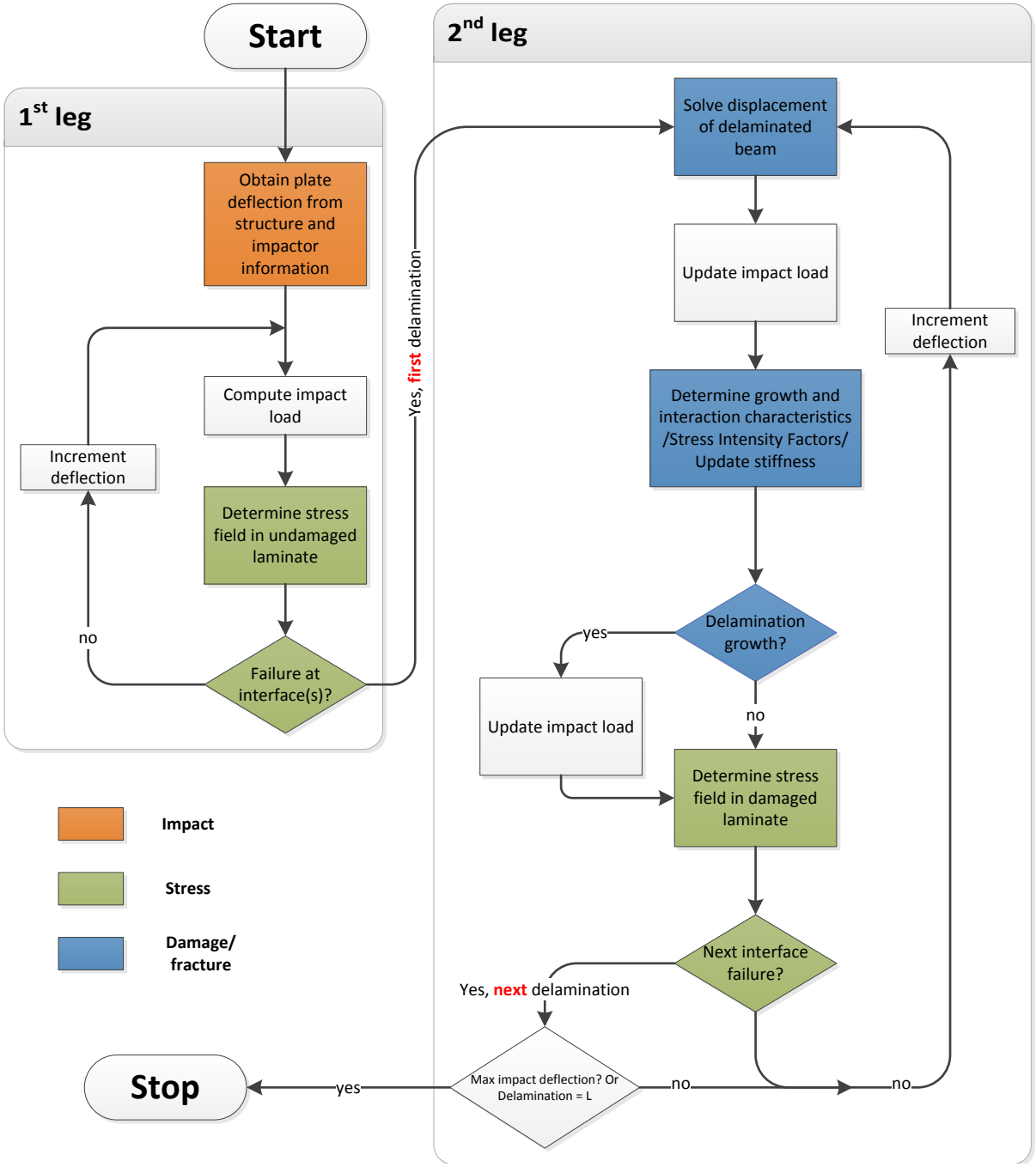


Figure 3.1 - Schematic overview of the proposed model

damage under a quasi-static impact as accurately as possible. The reason why it should be fast relates to the fact that such a tool could be used in an optimizing routine preselecting a range of laminates which show good damage resistance/tolerance characteristics. In developing this model a modular approach was chosen, allowing to take individual blocks and replace them with alternative more complex ones, resulting in a more accurate, but possibly more expensive, solution. As previously mentioned the current model revolves around three main modules: the impact module, the stress state module and the damage and fracture module. In Figure 3.1 these modules have been placed in a flow diagram, presenting the approach in a visual form while acting as a roadmap for this chapter.

Clearly these blocks represent the separate research tasks set up in section 1. As can be seen from Figure 3.1 the impact module discussed in section 3.3 only plays a minor role in the entire process since it will only have to be considered in the beginning. It will be used to determine whether or not the impact event on the given structure can be considered quasi-static using certain guidelines. Further, it is used to compute the deflection of the beam under impact and the maximum indentation due to the impactor. The stress determination in the undamaged laminate with subsequent application of a failure criterion is still rather straightforward and discussed in sections 3.4-3.6. The main difficulty therefore lies in the second leg of the flow diagram, starting with the damage/fracture module encompassing the analysis of one or multiple delaminations including their interaction and growth characteristics and the inclusion of damage in the stiffness of the structure (sections 3.7 to 3.9). Finally, determining the stress in the damaged laminate is paramount for knowing at which interface the next delamination will initiate. The new internal stress state is based on the stress field emanating from the delamination tip, the updated bending stresses and the contact stresses, and will be discussed in section 3.9. The governing assumption for each of the building blocks will be discussed in the respective sections.

3.2 Composite prerequisites

When dealing with composite mechanics it is helpful to clarify certain concepts beforehand regarding the constitutive parameters, and for some of the analysis done here transversely isotropic material properties have been assumed, which will be discussed hereafter. There are many textbooks which cover the basic principles of composite structures from the constitutive relationships of a single ply to the governing relationships of the entire laminate (Reddy [12] and Kassapoglou [4]).

Often it is useful to work with the engineering constants of a laminate. Those constants can be derived from the stiffness matrices of the laminate. For the case of a symmetric and balanced laminate the membrane stiffness in the principal direction of the laminate is given as:

$$E_{m,1} = \frac{1}{h a_{11}} \tag{3.2.1}$$

where a_{11} is the (1,1)-entry of the laminate compliance matrix, which in case of a symmetric and balanced laminate is simply the inverse of the in-plane stiffness matrix A, obtained from classical laminated plate theory (CLPT). Similarly for the transverse in-plane laminate direction we have the following expression:

$$E_{m,2} = \frac{1}{h a_{22}}$$

Below are those constants which are needed for the remainder of the project:

$$E_{m,1}=\frac{1}{ha_{11}};\ E_{m,2}=\frac{1}{ha_{22}};\ \nu_{m,12}=-\frac{a_{12}}{a_{11}};\ \nu_{m,21}=-\frac{a_{12}}{a_{22}} \tag{3.2.2}$$

with ν being the Poisson's ratio. As we will see later (section 3.5) it is often convenient to assume axially symmetric deflections when dealing with impact problems. This implicitly assumes that transverse shear stiffnesses and strengths are equal to each other, which is reasonable as the actual values do not differ too much from each other. This assumption can be applied to quasi-isotropic (QI) layups without too much care while for directional laminates it becomes more critical. Using axially symmetric deflections leads to the use of polar coordinates rather than cartesian. In this case engineering constants also exist, however, they are merely approximated [6]. Using equation (3.2.2) the following relationships can be written down:

$$\begin{aligned} E_r &= \frac{E_{m,1}+E_{m,2}}{2} \\ \nu_{r\theta} &= \frac{\nu_{m,12}+\nu_{m,21}}{2} \end{aligned} \tag{3.2.3}$$

R and θ are the in-plane coordinates. In order to be able to define the a transversely isotropic material we also need E_z, ν_{rz} and G_{rz}. Suemasu, Kerth [42] derived the following equations in their paper:

$$\begin{aligned} E_z &= A_{s,22}-\frac{\left(A_{s,12}+A_{s,23}\right)^2}{A_{s,11}+A_{s,22}+2A_{s,12}} \\ \nu_{rz} &= \frac{A_{s,12}+A_{s,23}}{A_{s,11}+A_{s,22}+2A_{s,12}} \\ G_{rz} &= 2G_{0,xy}\frac{\left(1-\nu_{0,13}\right)}{\left(2-\nu_{0,13}\right)} \end{aligned} \tag{3.2.4}$$

The coefficients $A_{S,ii}$ are defined as:

$$\begin{aligned} A_{s,11} &= \frac{E_{0,1}\left(1-\nu_{0,13}^2\right)}{\Delta_s};\ A_{s,22}=\frac{E_{0,2}\left(1-\nu_{0,12}\nu_{0,21}\right)}{\Delta_s}; \\ A_{s,12} &= \frac{E_{0,2}\nu_{0,12}\left(1+\nu_{0,13}\right)}{\Delta_s};\ A_{s,23}=\frac{E_{0,2}\left(\nu_{0,13}-\nu_{0,12}\nu_{0,21}\right)}{\Delta_s} \\ \Delta_s &= 1-2\nu_{0,12}^2\nu_{0,21}-\nu_{0,13}\nu_{0,12}-\nu_{0,13}^2 \end{aligned} \tag{3.2.5}$$

Here, all coefficients containing a 0 in their index are basic ply properties. These constants have been derived using a quasi-isotropic laminate made up of an infinite number of plies. Having defined these engineering constants we can now define the engineering compliance matrix for a transversely isotropic material according to Lekhnitskii [14]:

$$a_{11,hs}=a_{22,hs}=\frac{1}{E_r};\ a_{12,hs}=-\frac{\nu_{r\theta}}{E_r};\ a_{13,hs}=-\frac{\nu_{rz}}{E_z};\ a_{33,hs}=\frac{1}{E_z};\ a_{44,hs}=\frac{1}{G_{rz}} \tag{3.2.6}$$

These compliances are then used when defining constants during the derivation of the governing stress function for the transverse isotropic body. Those constants are given below and will be used in section 3.5:

$$a=\frac{a_{13,hs}(a_{11,hs}-a_{12,hs})}{\Delta};\ b=\frac{a_{13,hs}(a_{13,hs}+a_{44,hs})-a_{12,hs}a_{33,hs}}{\Delta}$$
$$c=\frac{a_{13,hs}(a_{11,hs}+a_{12,hs})+a_{11,hs}a_{44,hs}}{\Delta};\ d=\frac{a_{11,hs}^2-a_{12,hs}^2}{\Delta} \qquad (3.2.7)$$
$$\Delta=a_{11,hs}a_{33,hs}-a_{13,hs}^2$$

It has to be emphasized at this point that assuming any composite as transversely isotropic is of course very simplifying as composites normally feature anisotropic or orthotropic in-plane behaviour. However, this assumption allows for a variety of approaches to be applied and certainly is considered more accurate than an isotropic representation of the composite.

3.3 Quasi-static impact analysis

As indicated in Figure 3.1 the whole process starts with the analysis of the impact event. The impact event is primarily governed by the velocity and the mass of the impactor and can be categorized into three basic cases as shown in Figure 3.2. The impact response considered here is the one depicted in Figure 3.2c). It is characterized by the in-phase behaviour of impactor and structure [2], i.e. of the deflection of the target and the load acting on the plate. It results from the fact that the impact time is much longer than it takes for elastic waves to travel to the plate boundaries. Its validity can be evaluated by looking at the ratio of impactor mass and effective structural mass explained hereafter.

3.3.1 Assumptions

The main assumption that has to be made here is that of a quasi-static response, which will simplify the computations significantly. A rule-of-thumb for the applicability of this assumption implies that the impact frequency is less than a third of the lowest natural frequency occurring [11], or:

$$\omega_{imp}\leq\frac{1}{3}\omega_n \qquad (3.3.1)$$

where both ω_{imp} and ω_n can be approximated by:

$$\omega_{imp}\approx\sqrt{\frac{k_{eq}}{M_{imp}+m_{eq}}}\ \text{and}\ \omega_n\approx\sqrt{\frac{k_{eq}}{m_{eq}}} \qquad (3.3.2)$$

where k_{eq} is the equivalent stiffness of the target, M_{imp} is the mass of the impactor and m_{eq} the equivalent lumped mass of the target. Using equations (3.3.2) and (3.3.1) we can arrive at limit in terms of the involved masses for a quasi-static approximation of the impact problem:

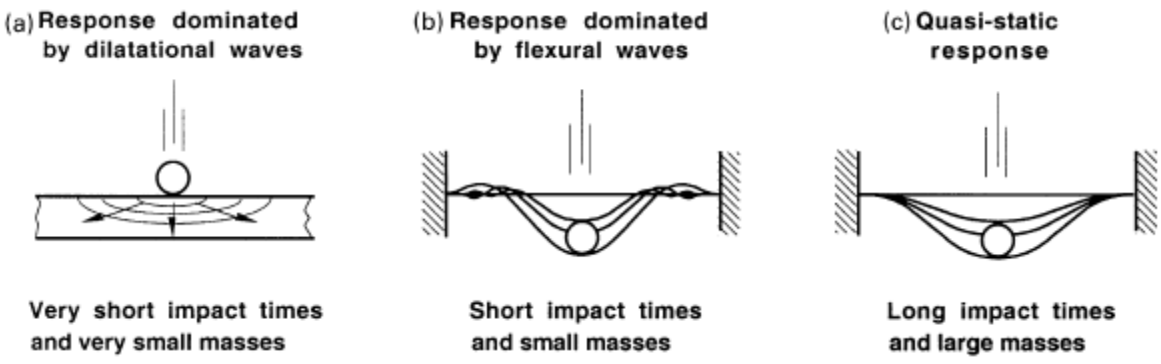


Figure 3.2 - Classification of response types. (Figure taken from Olsson [2])

$$M_{imp} \geq 8m_{eq} \tag{3.3.3}$$

This equivalent lumped mass can be obtained by using the implicit expression below:

$$\frac{1}{2}m_{eq}\delta_c^2 = \frac{1}{2}\int_{Vol} \rho w^2 dVol \tag{3.3.4}$$

where δ_c is the central deflection of the structure, ρ the density of the target material and w the static displacement of the beam. This implies that once the static displacement field due to the concentrated force is known the equivalent mass can be determined, which will depend primarily on the boundary conditions. For a beam either clamped or simply supported at both sides the ratios of equivalent to total mass are 0.371 and 0.486, respectively. If the mass of the target increases while the impactor mass stays constant flexural waves will play a more pronounced role and the behaviour will be similar to the one depicted in Figure 3.2 b).

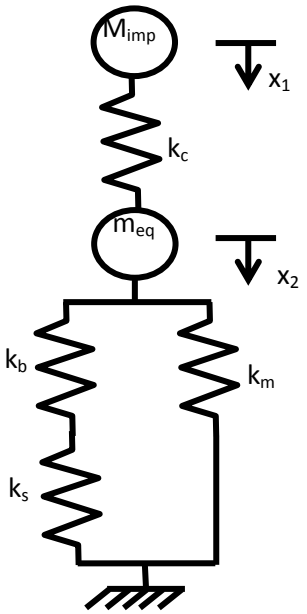


Figure 3.3 - Two degree of freedom system used for the impact analysis

Next to that we will further assume applied displacement as the input for our simulation. In reality the impact can neither be simulated by an applied force nor displacement approach as both cases happen simultaneously. Using a dynamic model one could potentially implement both approaches. The choice for the applied displacement approach stems from the fact that most experiments are carried out under stroke controlled conditions. Based on that a two degree of freedom (TDOF) equivalent spring system can be modelled as shown in Figure 3.3. In this system the spring stiffnesses k_c, k_b, k_s and k_m correspond to the contact, bending, shear and membrane stiffness respectively. If only small deflections are considered then the membrane action of the impacted structure can be neglected, which simplifies the system. The Hertzian formulation is assumed for the contact. The contact force can then be computed according to:

$$P_{con} = k_c \alpha^{3/2} \tag{3.3.5}$$

where α is the indentation defined as the difference between the displacement of the impactor and the central deflection of the plate, $\alpha = x_1 - x_2$. The contact stiffness between an isotropic impactor and a composite plate is given by:

$$k_c = \frac{4\sqrt{R_{imp}}}{3\pi(n_1 + n_2)} \tag{3.3.6}$$

where R_{imp} is the radius of the spherical impactor. Here n_1 and n_2 are impactor and target specific parameters and are defined as:

$$n_1 = \frac{1-\nu_{imp}^2}{\pi E_{imp}} \tag{3.3.7}$$

and

$$n_2 = \frac{\sqrt{C_{rr}}\sqrt{\left(\sqrt{C_{zz}C_{rr}} + G_{rz}\right)^2 - \left(C_{rz} + G_{rz}\right)^2}}{2\pi\sqrt{G_{rz}}\left(C_{zz}C_{rr} - C_{rz}^2\right)}$$

with

$$C_{zz} = \frac{E_z(1-\nu_{r\theta})}{\delta};\ C_{rr} = \frac{E_r(1-\nu_{rz}^2)}{(1+\nu_{r\theta})\delta};\ C_{rz} = \frac{E_r\nu_{rz}}{\delta}$$

$$\delta = 1 - \nu_{r\theta} - 2\nu_{rz}^2 \qquad (3.3.8)$$

The last equation has been published by Suemasu, Kerth [42] and is valid for transversely isotropic plates. The expression simplifies to the same form as given in (3.3.7) for an isotropic target. The constitutive parameters in the equation above have previously been defined in section 3.2.

While the mass ratio of structure and impactor is an important indicator for the analysis approach it also has to be mentioned up to which energy levels the simulation is plausible. Lopes' studies have shown that up to an incident energy of 20 J the form of damage is mainly due to delaminations and will therefore be used as an upper limit for the impact event subsequently. This does not mean that the model will not work beyond this energy level, however, the accuracy and validity of the obtained results are questionable.

3.3.2 Force and indentation during impact

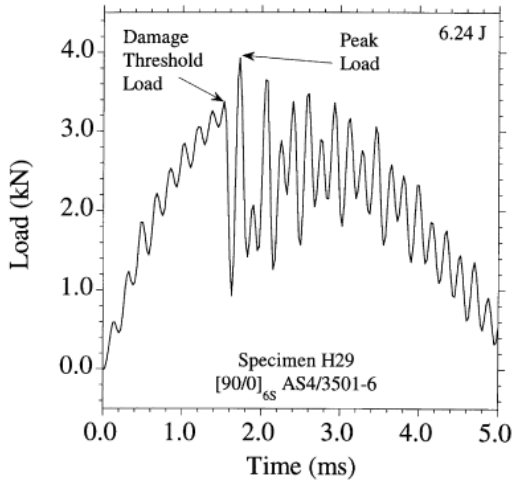


Figure 3.4 - exemplary force history during impact showing features of the delamination threshold load (Figure obtained from Schoeppner and Abrate [3])

Having determined the limit for the quasi-static response we can proceed to determine the load and indentation history based on the displacement of the beam under impact. It has been shown that for a quasi-static approximation the load history has a sinusoidal form [3]. However, if during an impact event internal damage occurs in the form of delaminations, it will have a noticeable effect on the load history in the form of an abrupt drop in the applied load (see Figure 3.4). This effect cannot be captured by a simple sinusoidal representation of the impact event. Nevertheless, we can still use this approach to obtain the maximum displacement that will be encountered during impact together with the maximum indentation. Alternatively, an energy balance can be set up with which similar results can be achieved.

The two equations of motions for the TDOF system as shown in Figure 3.3 are:

$$M_{imp}\frac{d^2x_1}{dt^2} + k_c\alpha^{3/2} = 0$$

$$m_{eq}\frac{d^2x_2}{dt^2} + k_{eq}x_2 - k_c\alpha^{3/2} = 0 \qquad (3.3.9)$$

Here we have already taken into account that $k_m = 0$. The equivalent stiffness k_{eq} is defined as:

$$k_{eq} = \frac{k_b k_s}{k_b + k_s} \qquad (3.3.10)$$

Where k_b and k_s are bending and shear stiffness of the beam, respectively. If we only consider non-shear deformable beams we have:

$$k_{eq} = k_b$$
$$\text{with } k_b = \frac{CE_{xx}^b I_{yy}}{L^3} \tag{3.3.11}$$

where I_{yy} and L are moment of inertia and length of the beam, respectively. In the expression for k_b the factor C represents the boundary conditions and is either 48 or 192 for either simply supported or clamped conditions. E_{xx}^b in expression (3.3.11) is based on lamination theory for beams and can be defined as [12]:

$$E_{xx}^b = \frac{12}{h^3 d_{11}} \tag{3.3.12}$$

Again, d_{11} is the (1,1)-entry of the compliance matrix d. We can now express the contact force as:

$$P_{con} = P_{con,m} \sin\left(\frac{\pi}{2t_m} t\right) \tag{3.3.13}$$

where t_m is the time until the force has reached its maximum. After some algebraic manipulation (see Pang, Zhao [9]) one can arrive at a single non-linear equation in terms of the maximum indentation α_m , which can solved by the Newton-Raphson method:

$$\frac{2m_{eq}k_c^3}{3M_{imp}k_{bs}v_0^2}\alpha_m^{11/2} - \frac{k_c^2}{k_{bs}}\left(1+\frac{m_{eq}}{M_{imp}}\right)\alpha_m^3 - k_c\alpha_m^{5/2} + M_{imp}v_0^2 = 0 \tag{3.3.14}$$

where v_0 is the impactor velocity at impact and the index m indicates the maximum value obtained during impact. The central deflection of the beam follows from:

$$w_c(t) = \frac{4P_{con,m}t_m^2}{\pi^2 M_{imp}} \sin\left(\frac{\pi}{2t_m} t\right) + \left(v_0 - \frac{2P_{con,m}t_m}{\pi m_{eq}}\right)t - \alpha_m \sin\left(\frac{\pi}{2t_m} t\right)^{2/3} \tag{3.3.15}$$

While the applied load for an undamaged beam follows from (3.3.13) this equation is no longer valid once damage has occurred. In section 3.9 we will discuss how this damage is used to update the applied load.

The alternative approach to get to displacement already mentioned relies on the fact that the incident energy from the impact is equal to the deformation energy of the structure consisting of contact, bending, shearing and membrane deformations in case of no energy dissipation due to the creation of damage:

$$E_{imp} = E_c + E_{bs} + E_m \tag{3.3.16}$$

Again, neglecting the energy due to membrane action and after some manipulation one can arrive at a non-linear relation in terms of the centre displacement w_c [8]:

$$k_{bs}w_c^2 + \frac{4}{5}\left(\frac{k_{bs}w_c^5}{k_c^2}\right)^{1/3} - M_{imp}v_0^2 = 0 \tag{3.3.17}$$

The maximum displacement w_c can be found through application of the Newton-Raphson method once more. Multiplication with the bending stiffness k_b then yields the applied load.

Either of these two approaches will render a maximum central beam deflection. The simulation will carry through until that displacement has been reached. The indentation found alongside can be used as an additional damage indicator for the top layer.

3.4 Global structural analysis

The load acting on the plate during an impact event was obtained in the previous section. This load will be used in this section to carry out a structural analysis of the impacted beam.

3.4.1 Assumptions

The main assumptions governing the global structural analysis of a composite laminated beam are listed below:

- The laminates considered are symmetric resulting in a decoupling effect between the in- and out-of-plane deformations.
- The CLPT for laminated beams is applied (laminated beam theory (LBT)). This implies that the assumptions concerning plane stress are active. Further, Myy = Mxy = 0 everywhere and the Poisson effect and the anisotropic shear coupling are assumed to be negligible [12]. The latter is equivalent to saying that $D_{16} = D_{26} = 0$.
- No membrane action or shear deformation is considered.
- The beam is either simply supported or clamped at both ends.

3.4.2 Stress analysis

Unless stated otherwise all equations are taken from [12]. The situation under consideration is shown in Figure 3.5. For the computation of the in-plane stresses we need knowledge of the internal bending moment distribution. For both cases this is readily available. The in-plane stresses per layer then can be written as:

$$\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{Bmatrix}^k = z\begin{bmatrix}\bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16}\\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26}\\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66}\end{bmatrix}^k \begin{Bmatrix}-\dfrac{\partial^2 w}{\partial x^2}\\ -\dfrac{\partial^2 w}{\partial y^2}\\ -2\dfrac{\partial^2 w}{\partial x \partial y}\end{Bmatrix} = z\begin{bmatrix}\bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16}\\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26}\\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66}\end{bmatrix}^k \begin{bmatrix}d_{11} & d_{12} & 0\\ d_{12} & d_{22} & 0\\ 0 & 0 & d_{66}\end{bmatrix} \begin{Bmatrix}M_{xx}\\ 0\\ 0\end{Bmatrix}$$

or

$$\sigma_{xx}^k(x,z) = \frac{M(x)z}{b}\left(\bar{Q}_{11}^k d_{11} + \bar{Q}_{12}^k d_{12}\right) \qquad (3.4.1)$$

$$\sigma_{yy}^k(x,z) = \frac{M(x)z}{b}\left(\bar{Q}_{12}^k d_{11} + \bar{Q}_{22}^k d_{12}\right)$$

$$\sigma_{xy}^k(x,z) = \frac{M(x)z}{b}\left(\bar{Q}_{16}^k d_{11} + \bar{Q}_{26}^k d_{12}\right)$$

In equation (3.4.1) $\bar{Q}_{ij}$ are the plane-stress reduced rotated ply stiffnesses and M_{xx} the applied moment per unit width. The matrix $\bar{Q}$ is obtained by taking the 3 by 3 plane-stress reduced stiffness matrix for an orthotropic ply, which is defined in the local ply coordinate system, and transform it to a global laminate coordinate system using tensor transformation. The superscript k denotes the kth layer of the laminate. In the second line of equation (3.4.1) M_{xx} has been replaced by the actual

moment M obtained from statics divided by the physical width of the beam b. In both cases the maximum bending moment occurs at mid-span and is equal to

$$M = \frac{PL}{4} + M_r \text{; with } M_r = 0 \text{ for simply supported beams} \tag{3.4.2}$$

The constant root moment M_r for the clamped beam has to be computed by solving the static problem of a beam containing several delaminations (see section 3.7)

As we are also interested in the transverse stress components we can obtain these from considering the three dimensional equilibrium equations:

$$\begin{aligned} 0 &= \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \\ 0 &= \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \\ 0 &= \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \end{aligned} \tag{3.4.3}$$

By rearranging these expressions and integrating with respect to z within each layer we obtain the relations for the stress components σ_{xz}, σ_{yz} and σ_{zz} :

$$\begin{aligned} \sigma_{xz}^k &= -\int_{z_k}^{z_{k+1}} \frac{\partial \sigma_{xx}^k}{\partial x} dz + F^k \\ \sigma_{yz}^k &= -\int_{z_k}^{z_{k+1}} \frac{\partial \sigma_{xy}^k}{\partial x} dz + G^k \\ \sigma_{zz}^k &= -\int_{z_k}^{z_{k+1}} \frac{\partial \sigma_{xz}^k}{\partial x} dz + H^k \end{aligned} \tag{3.4.4}$$

In deriving the expressions we have already taken into account that any derivatives with respect to

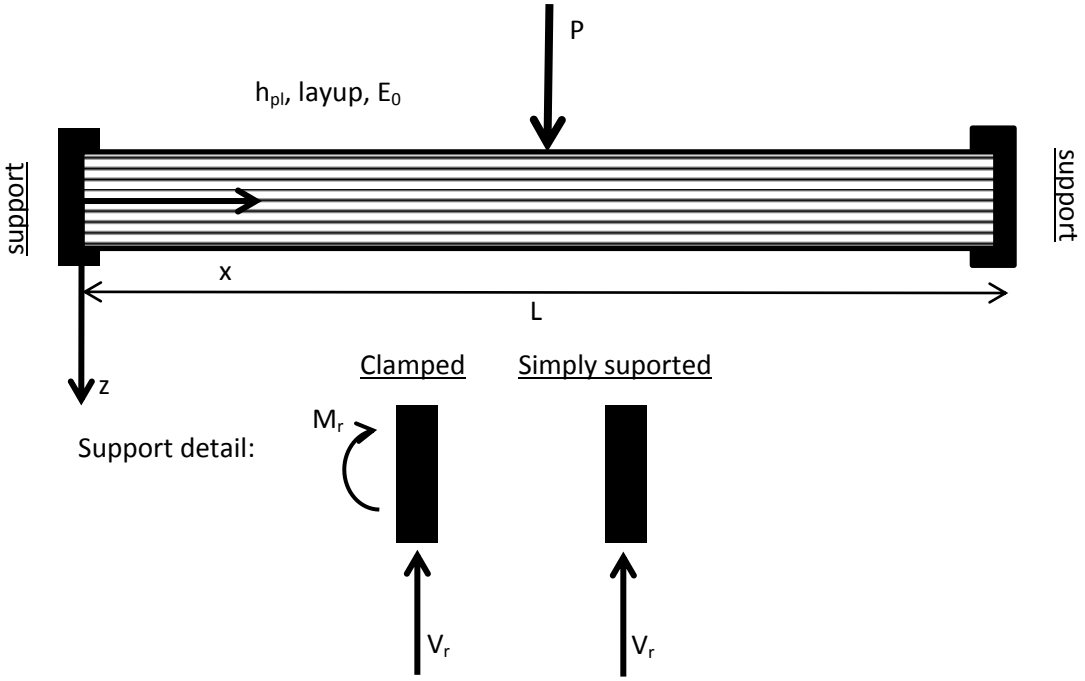


Figure 3.5 – Composite beam under a point load with either clamped or simply supported boundary conditions

y can be ignored in laminated beam theory. The only variable in dependence of x in the expressions for the in-plane stresses is the applied moment, which is why the expressions for the transverse shear stresses can be written as:

$$\sigma_{xz}^{k}(x,z) = -\frac{Q}{b}\left(\bar{Q}_{11}^{k} d_{11} + \bar{Q}_{12}^{k} d_{12}\right)\left(\frac{z^2 - z_k^2}{2}\right) + F^k$$
$$\sigma_{yz}^{k}(x,z) = -\frac{Q}{b}\left(\bar{Q}_{16}^{k} d_{11} + \bar{Q}_{26}^{k} d_{12}\right)\left(\frac{z^2 - z_k^2}{2}\right) + G^k \quad \text{with } Q = \frac{P}{2} \tag{3.4.5}$$

From these expressions we see that the transverse shear stresses vary piecewise quadratically through the thickness. The transverse normal stress σ_{zz} can be written as:

$$\sigma_{zz}^{k} = -\frac{dQ}{dx}\frac{1}{b}\left(\bar{Q}_{11}^{k} d_{11} + \bar{Q}_{12}^{k} d_{12}\right)\left(\frac{z^3 - z_k^3}{6}\right) + H^k \tag{3.4.6}$$

It has to be noted that in case of a concentrated point load the derivative of the shear force does not exist. The only way this stress component can be recovered would be to model the impact as distributed load acting over a finite impact radius, which will be discussed in the next section. The integration constants F^k, G^k and H^k can be obtained from the condition that those stresses need to be continuous across the ply interfaces:

$$\sigma_{iz}^{k}(x, z_{k+1}) = \sigma_{iz}^{k+1}(x, z_{k+1}), i = x, y, z \tag{3.4.7}$$

At the bottom of the laminate all stress components are zero, hence:

$$F^1 = 0$$
$$G^1 = 0 \tag{3.4.8}$$

All further constants are then defined as:

$$F^{k+1} = \sigma_{xz}^{k}(x, z_{k+1})$$
$$G^{k+1} = \sigma_{yz}^{k}(x, z_{k+1}) \tag{3.4.9}$$

With the exception of the transverse normal stress σ_{zz} for a concentrated load, it is possible to define five stress components in this way. However, the stresses due to the contact have to be taken into account which will be done in the following section.

3.5 Local contact analysis

Having obtained the stresses from a global structural analysis we proceed in this section to determine the contact stresses due to impact on a composite laminate. As mentioned before, an accurate knowledge of the stress state within the impacted structure is crucial to determine when and where a delamination will initiate. It is generally known, that during impact very high and localized contact stresses occur, which need to be accounted for.

A schematic overview of the situation under consideration is presented in Figure 3.6. Note that different authors have been using different coordinate systems, which merely causes the positioning of the z-axis to differ ([16], [15], [17] and [18]). When comparing the results, this has been taken care of. After stating the governing assumptions for determining the contact stresses, the solution for semi-infinite bodies will be presented first followed by the finite thickness formulation and finally a comparison between the two.

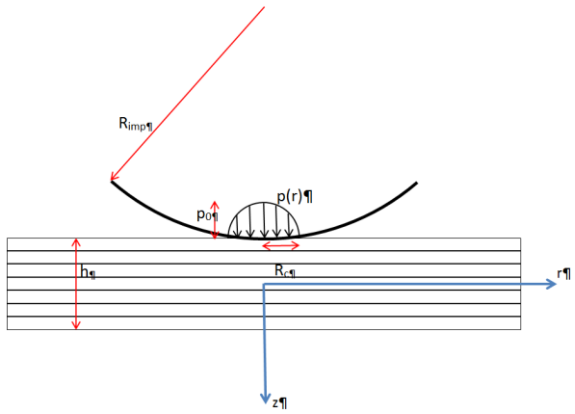


Figure 3.6 - schematic overview of the impact

3.5.1 Assumptions

For the computation of contact stresses we can generally consider either half-spaces or finite thickness plates. The advantage of working with half-spaces, which assume infinitely deep material, is that closed form expressions for both the transverse normal and shear stress exist. However, when dealing with thin plates these results become rather inaccurate. If we consider a finite thickness of a laminate, no closed form solution to the problem exists and numerical methods will have to be used to arrive at the stress distribution. Both approaches assume an infinitely wide plate and an axially symmetric problem set up, i.e. polar coordinates are used and all derivatives with respect to θ are zero. Therefore the only non-zero stress components are defined in terms of a stress function as:

$$\begin{aligned}
\sigma_{rr} &= -\frac{\partial}{\partial z}\left(\frac{\partial^2\psi}{\partial r^2}+\frac{b}{r}\frac{\partial\psi}{\partial r}+a\frac{\partial^2\psi}{\partial z^2}\right)\\
\sigma_{\theta\theta} &= -\frac{\partial}{\partial z}\left(b\frac{\partial^2\psi}{\partial r^2}+\frac{1}{r}\frac{\partial\psi}{\partial r}+a\frac{\partial^2\psi}{\partial z^2}\right)\\
\sigma_{zz} &= \frac{\partial}{\partial z}\left(c\frac{\partial^2\psi}{\partial r^2}+\frac{c}{r}\frac{\partial\psi}{\partial r}+d\frac{\partial^2\psi}{\partial z^2}\right)\\
\sigma_{rz} &= \frac{\partial}{\partial r}\left(\frac{\partial^2\psi}{\partial r^2}+\frac{1}{r}\frac{\partial\psi}{\partial r}+a\frac{\partial^2\psi}{\partial z^2}\right)
\end{aligned} \tag{3.5.1}$$

where the coefficients a-d have been defined in (3.2.7). These expressions naturally simplify when an isotropic material is considered. The main difference between the two approaches lies in the formulation of the boundary conditions when solving for the stress function Ψ. In the finite thickness approach two constitutive relationships are considered, isotropic and transversely isotropic. For the transverse normal stress σ_{zz} and the transverse shear stress σ_{rz} there is one particular radial position at which the stresses through the thickness are highest. For the normal stress this position is at r=0 and for the shear stress this position is at $r=R_c$. However, for the failure of a particular interface any position in between can be equally important due to the combined action of transverse shear and normal stresses. Therefore these stresses need to be computed for a range of values for z and r.

3.5.2 Semi-infinite thickness approach

The underlying assumption of this approach specifies all stress components will vanish if the z coordinate goes to infinity. Clearly, this is an assumption, which depending on the actual thickness of the structure one analyses, can be too simplifying. Nevertheless, this approach has been used extensively and almost all of the expressions for the stress components are given in closed-form which results in a quick assessment of the situation [16]. The expressions used have been taken from [15]. They have solved the problem of the elastic contact between a sphere and a transversely isotropic half-space based on Lekhnitskii's formulation of anisotropic elasticity. The complete

derivation can be found in their paper and only the final expressions will be given here. The transverse normal and shear stresses are given as:

$$\begin{aligned}
\sigma_{rr} &= p_0\left\{\frac{-1}{(s_1-s_2)\sqrt{d}}\left(s_1C_{1,2}-s_2C_{2,2}\right)+\frac{\nu}{r(s_1-s_2)}\left(s_1\rho_2D_{1,3}-s_2\rho_1D_{2,3}\right)\right\}\\
\sigma_{\theta\theta} &= p_0\left\{\frac{\sqrt{d}}{(s_1-s_2)(ac-d)}\left(s_1q_2C_{1,2}-s_2q_1C_{2,2}\right)+\frac{\nu}{r(s_1-s_2)}\left(s_1\rho_2D_{1,3}-s_2\rho_1D_{2,3}\right)\right\}\\
\sigma_{zz} &= p_0\frac{s_2C_{1,2}-s_1C_{2,2}}{s_1-s_2}\\
\sigma_{rz} &= p_0\frac{D_{1,2}-D_{2,2}}{(s_1-s_2)\sqrt{d}}
\end{aligned} \tag{3.5.2}$$

with:

$$\begin{aligned}
C_{i,2} &= 1-\frac{1}{R_c}\left(rS^1_{i,1}+s_izS_{i,1}\right)\\
D_{i,2} &= \frac{1}{2R_c}\left(rS_{i,1}-s_izS^1_{i,1}-R_cT^1_{i,1}\right)\\
D_{i,3} &= \frac{1}{3R_c}\left(R_c^2-r^2+\frac{s_i^2z^2}{2}\right)S^1_{i,1}+\frac{s_1z}{6}T^1_{i,1}-\frac{rs_iz}{2R_c}S_{i,1}+\frac{r}{3}\\
S_{i,1} &= \tan^{-1}\left(\frac{R_c-\beta_i}{\alpha_i+s_iz}\right);\ S^1_{i,1}=\frac{\beta_i+R_c}{r};\ T^1_{i,1}=\frac{\alpha_i-s_iz}{r}\\
\gamma_i^4 &= \frac{1}{R_c^4}\left(r^2-R_c^2+s_i^2z^2\right)^2+4\frac{s_i^2z^2}{R_c^2};\ \alpha_i^2=\frac{1}{2}\left(r^2+s_i^2z^2-R_c^2+R_c^2\gamma_i^2\right);\ \beta_i=-\frac{s_izR_c}{\alpha_i}\\
\rho_i &= 1-as_i^2;\ q_1=\left(b-as_2^2\right)\rho_1;\ q_2=\left(b-as_1^2\right)\rho_2
\end{aligned} \tag{3.5.3}$$

In equation (3.5.3) s_1 and s_2 represent the solution to the fourth order differential equation in an axially symmetric coordinate system and are defined as:

$$s_{1,2}=\sqrt{\frac{a+c\pm\sqrt{(a+c)^2-4d}}{2d}} \tag{3.5.4}$$

where the coefficients a-d have been defined previously in section 3.2. The pressure distribution as shown in Figure 3.6 is given by:

$$p(r)=\frac{p_0}{R_c}\sqrt{R_c^2-r^2} \tag{3.5.5}$$

with p_0 and R_c :

$$p_0=\frac{3P}{2\pi R_c^2};\ R_c=\left[\frac{3\pi}{4}PR_{imp}\left(n_1+n_2\right)\right]^{1/3} \tag{3.5.6}$$

In equation (3.5.6) n_1 and n_2 have previously been defined in equations (3.3.7) and (3.3.8). Simplifications of the expressions for σ_{zz} and σ_{rz} are possible for special locations. For the normal stress along z at r=0 we get:

$$\sigma_{zz} = -p_0 \left\{ 1 + \frac{s_1 s_2}{s_1 - s_2} \frac{z}{R_c} \left[\tan^{-1}\left(\frac{R_c}{s_1 z} \right) - \tan^{-1}\left(\frac{R_c}{s_2 z} \right) \right] \right\} \tag{3.5.7}$$

Simplifying the expression for σ_{rz} at $r=R_c$ yields:

$$\sigma_{rz} = \frac{p_0}{(s_1 - s_2)\sqrt{d}} \left\{ \frac{1}{2} \left[\tan^{-1}\left(\frac{R_c}{s_1 z R_1} \right) - \tan^{-1}\left(\frac{R_c}{s_2 z R_2} \right) \right] + \frac{z}{R_c} \left[s_2 \left(\frac{R_2^2 - 1}{R_2} \right) - s_1 \left(\frac{R_1^2 - 1}{R_1} \right) \right] \right\}$$
$$\text{with } R_i = \sqrt{\frac{1}{2}\left(1 + \sqrt{1 + 4\frac{R_c^2}{s_i^2 z^2}} \right)} \tag{3.5.8}$$

Equations (3.5.7) and (3.5.8) will be used in the comparison against the finite-thickness approaches later on.

3.5.3 Finite thickness approach

In contrast to the half-space problem there exists no closed-form solutions as yet for the contact stresses when the finite thickness of the plate is considered. The governing equation for the stress function ψ can normally be rendered in a biharmonic equation. Solving this biharmonic equation yields four unknowns, two of which can be cancelled when conditions at infinity are considered as one part of the boundary. However, if we regard a finite thickness system, all four unknown coefficients need to be solved, resulting in a 4x4 matrix system (see section 3.5.3.1). Due to the complexity of the expressions involved a need for numerical integration is born. In what follows, two different ways of how to determine the transverse stresses in an impacted composite are presented.

3.5.3.1 Solution involving Boussinesq's equation

In his paper, Talagani [18] combines the equations derived by Boussinesq with the Hankel transformation to arrive at two integral expressions for the transverse normal and shear stress. Boussinesq's equation are given as:

$$\begin{aligned} \nabla^2 B_0 &= 0 \\ \nabla^2 B_z &= 0 \\ u_r &= -\frac{1}{4(1-\nu_p)} \frac{\partial}{\partial r}\left(B_0 + z B_z \right) \\ u_z &= B_z - \frac{1}{4(1-\nu_p)} \frac{\partial}{\partial z}\left(B_0 + z B_z \right) \end{aligned} \tag{3.5.9}$$

The out-of-plane stresses for an axially symmetrical problem can be written in terms of the displacements as:

$$\begin{aligned} \sigma_{zz} &= \lambda\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} \right) + 2\mu \frac{\partial u_z}{\partial z} \\ \sigma_{rz} &= \mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \end{aligned} \tag{3.5.10}$$

Here λ and μ are the first and second Lamé parameters used to define isotropic material. Inserting (3.5.9) into (3.5.10) and rearranging the expressions we arrive at:

$$\sigma_{zz} = \frac{1-2\nu_p}{2(1-\nu_p)}(\lambda+2\mu)\frac{\partial B_z}{\partial z} - \frac{\mu}{2(1-\nu_p)}\left(\frac{\partial^2 B_0}{\partial z^2} + z\frac{\partial^2 B_z}{\partial z^2}\right)$$
$$\sigma_{rz} = \mu\left[\frac{1-2\nu_p}{2(1-\nu_p)}\frac{\partial B_z}{\partial r} - \frac{1}{2(1-\nu_p)}\left(\frac{\partial^2 B_0}{\partial r\partial z} + z\frac{\partial^2 B_z}{\partial r\partial z}\right)\right] \quad (3.5.11)$$

Next the Hankel transformation is defined as:

$$F(\zeta,z) = \int_0^\infty f(r,z) r J_0(\zeta r)\, dr$$
$$f(r,z) = \int_0^\infty F(r,z)\zeta J_0(\zeta r)\, d\zeta \quad (3.5.12)$$

where $J_0(\zeta r)$ is the zeroth order Bessel function of the first kind. This transformation can now be used to solve the partial differential equations in (3.5.9) and rewrite them as ordinary differential equations in the Hankel space, for which the solution is known. The solution for the functions in the r-z space can then be written as:

$$B_0(r,z) = \int_0^\infty \left(C_1(\zeta)e^{\zeta z} + C_2(\zeta)e^{-\zeta z}\right)\zeta J_0(\zeta r)\, d\zeta$$
$$B_z(r,z) = \int_0^\infty \left(C_3(\zeta)e^{\zeta z} + C_4(\zeta)e^{-\zeta z}\right)\zeta J_0(\zeta r)\, d\zeta \quad (3.5.13)$$

Finally we obtain the expressions for σ_z and τ_{rz} by inserting (3.5.13) into (3.5.11), resulting in:

$$\sigma_{zz} = \int_0^\infty \left\{\frac{1-2\nu_p}{2(1-\nu_p)}(\lambda+2\mu)\left(C_3 e^{\zeta z} - C_4 e^{-\zeta z}\right)\zeta - \frac{\mu\zeta^2}{2(1-\nu_p)}\left[C_1 e^{\zeta z} + C_2 e^{-\zeta z} + z\left(C_3 e^{\zeta z} + C_4 e^{-\zeta z}\right)\right]\right\}\zeta\, d\zeta$$
$$\sigma_{rz} = \mu\int_0^\infty \left\{-\frac{1-2\nu_p}{2(1-\nu_p)}\left(C_3 e^{\zeta z} + C_4 e^{-\zeta z}\right)\zeta + \frac{\zeta^2}{2(1-\nu_p)}\left[C_1 e^{\zeta z} - C_2 e^{-\zeta z} + z\left(C_3 e^{\zeta z} - C_4 e^{-\zeta z}\right)\right]\right\}\zeta J_1(R_c\zeta)\, d\zeta \quad (3.5.14)$$

where λ and μ are defined as:

$$\lambda = \frac{E_p\nu_p}{(1+\nu_p)(1-2\nu_p)};\ \mu = \frac{E_p}{2(1+\nu_p)} \quad (3.5.15)$$

The values for E_p and ν_p are taken from equation (3.2.3). It has to be stressed at this point that this approach is only valid for isotropic material as the constitutive modelling is solely based on the two parameters λ and μ. Using the expressions in (3.2.3) we homogenize the composite material and represent it as equivalently isotropic. It will be shown later how they compare to those derived for transversely isotropic materials. The coefficients C_1 to C_4 are determined by imposing that the shear stresses are zero at top and bottom surface and the normal stresses are zero at the bottom surface and follow a certain distribution on the impact site. The boundary conditions can therefore be modelled as:

$$\sigma_{zz}\left(r,\frac{h}{2}\right) = \begin{cases} p(r), & 0<r<R_c \\ 0 & \end{cases};\ \sigma_{zz}\left(r,-\frac{h}{2}\right) = 0$$
$$\sigma_{rz}\left(r,\frac{h}{2}\right) = \sigma_{rz}\left(r,-\frac{h}{2}\right) = 0 \quad (3.5.16)$$

With p(r) the pressure distribution at the impact site as given above (see Equation (3.5.5)). Since the integration is in terms of the transformed variable ζ we have to transform the pressure distribution as well, which results in:

$$\hat{p}(\zeta)=\int_0^{\infty} p(r)rJ_0(\zeta r)dr=\int_0^{R_c} p(r)rJ_0(\zeta r)dr=p_0\frac{\left(\sin(R_c\zeta)-\cos(R_c\zeta)\zeta R_c\right)}{\zeta^3 R_c} \qquad (3.5.17)$$

Combining (3.5.14), (3.5.16) and (3.5.17) results in the following matrix equation:

$$\begin{bmatrix} -\frac{\mu_p\zeta^2}{2(1-\nu_p)}e^{\zeta\frac{h}{2}} & -\frac{\mu_p\zeta^2}{2(1-\nu_p)}e^{-\zeta\frac{h}{2}} & \left[\frac{1-2\nu_p}{2(1-\nu_p)}(\lambda_p+2\mu_p)\zeta-\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}(\lambda_p+2\mu_p)\zeta-\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{-\zeta\frac{h}{2}} \\ -\frac{\mu_p\zeta^2}{2(1-\nu_p)}e^{-\zeta\frac{h}{2}} & -\frac{\mu_p\zeta^2}{2(1-\nu_p)}e^{\zeta\frac{h}{2}} & \left[\frac{1-2\nu_p}{2(1-\nu_p)}(\lambda_p+2\mu_p)\zeta+\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{-\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}(\lambda_p+2\mu_p)\zeta+\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{\zeta\frac{h}{2}} \\ \frac{\zeta^2}{2(1-\nu_p)}e^{\zeta\frac{h}{2}} & -\frac{\zeta^2}{2(1-\nu_p)}e^{-\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}\zeta+\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}\zeta-\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{-\zeta\frac{h}{2}} \\ \frac{\zeta^2}{2(1-\nu_p)}e^{-\zeta\frac{h}{2}} & -\frac{\zeta^2}{2(1-\nu_p)}e^{\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}\zeta-\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{-\zeta\frac{h}{2}} & \left[-\frac{1-2\nu_p}{2(1-\nu_p)}\zeta+\frac{\mu_p\zeta^2}{2(1-\nu_p)}\frac{h}{2}\right]e^{\zeta\frac{h}{2}} \end{bmatrix} \begin{bmatrix} C_1(\zeta) \\ C_2(\zeta) \\ C_3(\zeta) \\ C_4(\zeta) \end{bmatrix} = \begin{bmatrix} \hat{p}(\zeta) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(3.5.18)

This 4x4 system of linear equations has to be solved simultaneously for every ζ to render the four coefficients. Once these coefficients are obtained the stresses can be computed by numerical integration of expression (3.5.14). However, these integrals have “infinity” as the upper limit which poses a problem in terms of defining a reasonable limit for the numerical computation. Certainly there will be an end to the computations imposed by the limits of the computer the program is run on, since the computation of the determinant for high values of ζ yields unimaginable high values for the determinant. In fact, if nothing is changed in the settings of MATLAB then the program will return “infinity” after some time as a result, which automatically stops the computation. The behaviour of the integrand of the expressions in (3.5.14) is sinusoidal in nature with exponentially decaying amplitudes. The limit of the integration can therefore be determined by deciding when the integrand does not contribute noticingly towards the integrated stress value at given position z. It can be observed that the integrands are faster decaying as the computation marches through z

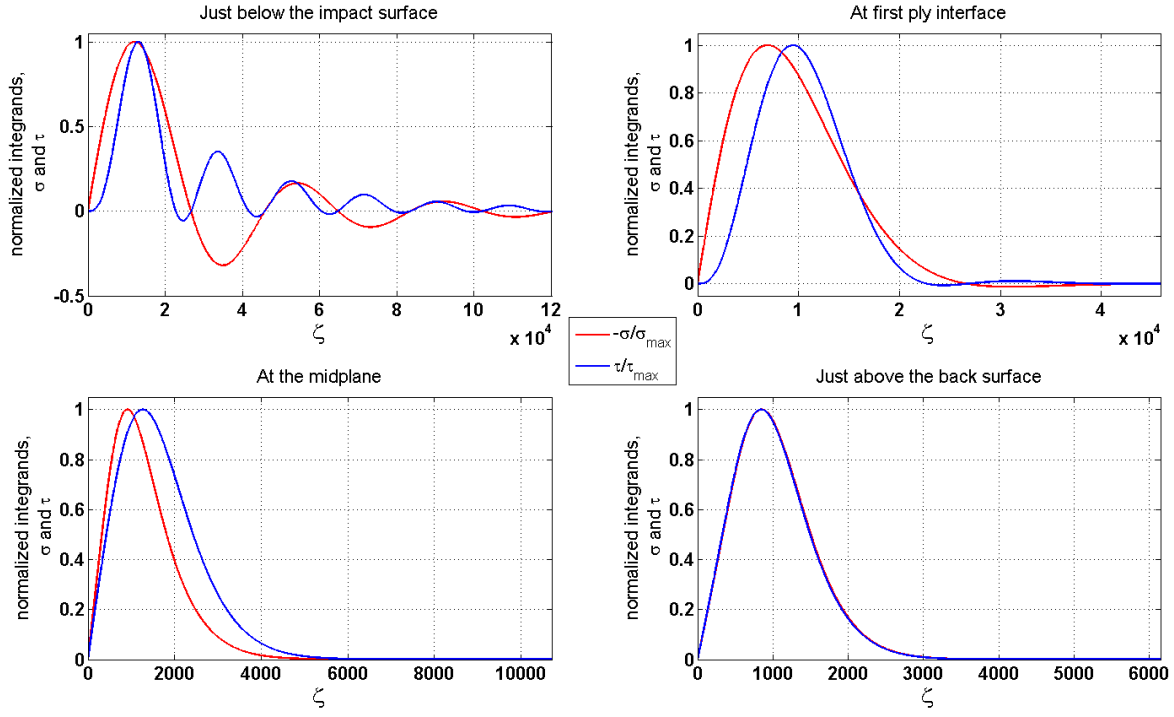


Figure 3.7 - Integrand plots at different thickness location

resulting in smaller ζ values for the integration limit closer to the back face of the laminate (see Figure 3.7). Note, in Figure 3.7 σ_{rz} has been replaced with τ while σ_{zz} is simply σ. Obviously, at the top surface the integration would stop once the value of the prescribed boundary value for σ_{zz} has been reached by numerical integration. This value can then be used as limit for the subsequent steps in z. The integrands shown in Figure 3.7 are taken at r=0 and r=R_c for σ_{zz} and σ_{rz}, respectively.

3.5.3.2 Solution used by Cairns

In his PhD thesis Cairns [17] presented a slightly different way of solving the contact problem. He took Lekhnitskii's formulation and imposed conditions at the impact face and the back face of the laminate, therefore losing the possibility to represent the solution in a closed form. Rather than integrating until infinity he defines the unknown stress function Ψ as infinite series according to:

$$\begin{gathered}\psi = \sum_{m=1}^{\infty} f_m(z) g_m(r) \\ g_m(r) = J_0(\omega_m r);\ \omega_m = \frac{\mu_m}{R_p};\ J_0(\mu_m) = 0 \\ f_m(z) = A_m e^{s_1 \omega_m z} + B_m e^{s_2 \omega_m z} + C_m e^{-s_1 \omega_m z} + D_m e^{-s_2 \omega_m z}\end{gathered} \tag{3.5.19}$$

Here J_0 is again the zeroth order Bessel function of the first kind. Once more the four coefficients have to be solved for every m by imposing the same boundary conditions as above (see (3.5.16)).

Inserting (3.5.19) into (3.5.1) we obtain:

$$\begin{aligned}\sigma_{rr} &= \sum_{m=1}^{\infty} -c g_m'' f_m' - \frac{b}{r} g_m' f_m' - a g_m f_m''' \\ \sigma_{\theta\theta} &= \sum_{m=1}^{\infty} -b g_m'' f_m' - \frac{1}{r} g_m' f_m' - a g_m f_m''' \\ \sigma_{zz} &= \sum_{m=1}^{\infty} c g_m'' f_m' + \frac{c}{r} g_m' f_m' + d g_m f_m''' \\ \sigma_{rz} &= \sum_{m=1}^{\infty} g_m''' f_m + \frac{1}{r} g_m'' f_m - \frac{1}{r^2} g_m' f_m + a g_m' f_m''\end{aligned} \tag{3.5.20}$$

where the apostrophe symbol denotes a derivative. The derivatives wrt r are given by:

$$\begin{aligned}g_m' &= -\omega_m J_1(\omega_m r) \\ g_m'' &= -\omega_m^2 J_0(\omega_m r) + \frac{\omega_m}{r} J_1(\omega_m r) \\ g_m''' &= \omega_m^3 J_1(\omega_m r) + \frac{1}{r} \omega_m^2 J_0(\omega_m r) - \frac{2}{r^2} \omega_m J_1(\omega_m r)\end{aligned} \tag{3.5.21}$$

whereas for z they are:

$$\begin{aligned}f_m' &= \omega_m \left(s_1 A_m e^{s_1 \omega_m z} + s_2 B_m e^{s_2 \omega_m z} - s_1 C_m e^{-s_1 \omega_m z} - s_2 D_m e^{-s_2 \omega_m z} \right) \\ f_m'' &= \omega_m^2 \left(s_1^2 A_m e^{s_1 \omega_m z} + s_2^2 B_m e^{s_2 \omega_m z} + s_1^2 C_m e^{-s_1 \omega_m z} + s_2^2 D_m e^{-s_2 \omega_m z} \right) \\ f_m''' &= \omega_m^3 \left(s_1^3 A_m e^{s_1 \omega_m z} + s_2^3 B_m e^{s_2 \omega_m z} - s_1^3 C_m e^{-s_1 \omega_m z} - s_2^3 D_m e^{-s_2 \omega_m z} \right)\end{aligned} \tag{3.5.22}$$

Inserting (3.5.22) and (3.5.21) into (3.5.20) results in the following expressions for σ_{zz} and σ_{rz}, which are necessary for solving the system:

$$\begin{aligned}\sigma_{zz} &= \sum_{m=1}^{\infty} J_0(\omega_m r)\omega_m^3 \left[\begin{array}{l} -c\left(s_1 A_m e^{s_1\omega_m z} + s_2 B_m e^{s_2\omega_m z} - s_1 C_m e^{-s_1\omega_m z} - s_2 D_m e^{-s_2\omega_m z}\right) + \\ d\left(s_1^3 A_m e^{s_1\omega_m z} + s_2^3 B_m e^{s_2\omega_m z} - s_1^3 C_m e^{-s_1\omega_m z} - s_2^3 D_m e^{-s_2\omega_m z}\right) \end{array} \right] \\ \sigma_{rz} &= \sum_{m=1}^{\infty} J_1(\omega_m r)\omega_m^3 \left[\begin{array}{l} \left(A_m e^{s_1\omega_m z} + B_m e^{s_2\omega_m z} + C_m e^{-s_1\omega_m z} + D_m e^{-s_2\omega_m z}\right) - \\ a\left(s_1^2 A_m e^{s_1\omega_m z} + s_2^2 B_m e^{s_2\omega_m z} + s_1^2 C_m e^{-s_1\omega_m z} + s_2^2 D_m e^{-s_2\omega_m z}\right) \end{array} \right] \end{aligned} \quad (3.5.23)$$

In order to match the coefficients A_m-D_m the pressure distribution has been modelled as:

$$\begin{aligned}\hat{p}(r) &= \sum_{m=1}^{\infty} \beta_m J_0(\omega_m r) \\ \beta_m &= \frac{2}{J_1^2(\mu_m) R_p^2} \int_0^{R_c} p(r) r J_0(\omega_m r) dr = \frac{2p_0}{J_1^2(\mu_m) R_p^2} \frac{\left(\sin(R_c\omega_m) - \cos(R_c\omega_m)\omega_m R_c\right)}{\omega_m^3 R_c}\end{aligned} \quad (3.5.24)$$

R_p is the radius of the plate affecting the solution, which has been taken as several times the contact radius in order to adapt to the increase in loading during the simulation. Combining (3.5.24) , (3.5.23) and (3.5.16) results in the following system of equations:

$$\omega_m^3 \begin{bmatrix} \left(-cs_1 + ds_1^3\right)e^{\omega_m s_1 \frac{h}{2}} & \left(-cs_2 + ds_2^3\right)e^{\omega_m s_2 \frac{h}{2}} & \left(cs_1 - ds_1^3\right)e^{-\omega_m s_1 \frac{h}{2}} & \left(cs_1 - ds_1^3\right)e^{-\omega_m s_2 \frac{h}{2}} \\ \left(-cs_1 + ds_1^3\right)e^{-\omega_m s_1 \frac{h}{2}} & \left(-cs_2 + ds_2^3\right)e^{-\omega_m s_2 \frac{h}{2}} & \left(cs_1 + ds_1^3\right)e^{\omega_m s_1 \frac{h}{2}} & \left(cs_2 - ds_2^3\right)e^{\omega_m s_2 \frac{h}{2}} \\ \left(1 - as_1^2\right)e^{\omega_m s_1 \frac{h}{2}} & \left(1 - as_2^2\right)e^{\omega_m s_2 \frac{h}{2}} & \left(1 - as_1^2\right)e^{-\omega_m s_1 \frac{h}{2}} & \left(1 - as_2^2\right)e^{-\omega_m s_2 \frac{h}{2}} \\ \left(1 - as_1^2\right)e^{-\omega_m s_1 \frac{h}{2}} & \left(1 - as_2^2\right)e^{-\omega_m s_2 \frac{h}{2}} & \left(1 - as_1^2\right)e^{\omega_m s_1 \frac{h}{2}} & \left(1 - as_2^2\right)e^{\omega_m s_2 \frac{h}{2}} \end{bmatrix} \begin{bmatrix} A_m \\ B_m \\ C_m \\ D_m \end{bmatrix} = \begin{bmatrix} \beta_m \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.5.25)$$

This 4x4 system can now be solved at every m and the stresses subsequently summed to give the final value. Similarly to the first method, the question of the upper limit of the sum arises. How many terms should be included? The value of the normal stress at the impact surface is known and equal to p_0. The summation can be carried out until the computed value at that point equals the value of p_0. The resulting value of m can then be used as an upper limit for the remainder of the z-coordinates.

3.5.4 Comparison of the two approaches

In this section we shall compare the results of the different approaches to finding the contact stresses near the area of impact. As we have seen above the only influence the actual layup of a laminate could have on the contact stresses is via the two parameters E_r and $\nu_{r\theta}$. For the comparison we will use two quasi-isotropic layups, namely $[0/45/90/-45]_{ns}$, where n is chosen between 1 and 3. We will also use two materials system with different degrees of orthotropy, defined as the ratio of the Young's modulus in longitudinal and transverse fibre direction. As already discussed the solution using Boussinesq's equations represents a homogenized isotropic material. In order to be able to compare this solution to a semi-infinite counterpart the approach developed by Dahan and Zarka [15] was used. In a limit process the solution to isotropic material can be found if s_1 and s_2 are approached to be 1. This solution is termed "DZ iso" in Figure 3.8. Since the plate only has two constitutive parameters the isotropic version of n_2 as defined in (3.3.8) has to be used resulting in a different contact pressure $p_{0,iso}$. The transverse normal stress σ_{zz} and the transverse shear stress σ_{rz} have been computed at r=0 and r=R_c, respectively. The ply properties are

given in Table 3.1. These values are typical for the material systems they represent. The properties for the steel impactor and the impact are also given in Table 3.1.

Table 3.1 - Parameters for the comparison study

Composite material	$E_{0,1}$ [GPa]	$E_{0,2}$ [GPa]	$G_{0,12}$ [GPa]	$\nu_{0,12}$ [-]	$\nu_{0,13}$ [-]	$E_{0,1}/E_{0,2}$ [-]
Glass/Epoxy[1] M1	60.7	24.8	12	0.23	0.34[2]	2.45
Graphite/Epoxy[3] M2	130.8	10.6	6.0	0.28	0.34	12.34

Impact material	E [GPa]	ν [-]	R [mm]	M [kg]	v_0 [m/s]
Steel	197	0.27	8	0.95	3.244

We computed the maximum impact load on a simply supported beam for every laminate and used this value as input for the stress determination. It has been found that with low values for P the difference between the two approaches is less noticeable. This is why the comparison was carried out at the maximum load level, as this is the worst case that can be expected. The graphs have been normalized with the contact pressure $p_{0,iso}$ in the case of σ_{zz} and with the highest occurring shear stress value for σ_{rz}. In Figure 3.8 a) and b) the two different material systems for n=1 are compared while in Figure 3.8 b)-d) the cases n=1..3 for material M2 are compared. It can be seen that in all cases the isotropic version predicts about twice as high stress values for both stress components. This is due to the fact that when approaching the plate as an isotropic material it is attributed a stiffer Young's modulus which results in a smaller contact radius for a given load and therefore a higher contact pressure p_0. A comparison of Figure 3.8 a) and b) shows, not surprisingly, that for a material with a lower degree of orthotropy this difference is less pronounced. Considering Figure 3.8 b)-c) one can clearly see that the finite thickness approaches of Cairns and Talagani succeed in recovering the boundary condition at the back face of the laminate whereas the semi-infinite formulations show a clear residual value for both stress components. It can be observed that this residual values decreases with increasing laminate thickness. This is regardless of the fact that the maximum load increases with increasing thickness. It can further be said that an almost imperceptible difference between the two approaches exist for both the isotropic and the transversely isotropic version of σ_{zz} up until the mid-plane of the laminate. This cannot be claimed for the transversely isotropic shear stress component, not even for large values of n. Here the differences, especially within the first few interfaces are significant and might tip the scales when it comes to applying failure criteria to determine delamination initiation. Keeping in mind that the entire model should be able to be executed quickly with respect to initial design selection, choosing the semi-infinite approach might prove to be the preferred method as it provides an instantaneous analysis with conservative maximum shear stress predictions. The fact that the transversely isotropic version differs so much from the isotropic version has been explained. Although by no means perfect, approaching a composite plate with a transversely isotropic material behaviour will certainly provide more reasonable results, which is reflected in the literature reviewed.

Before the solution for the contact stresses can be used in combination with the bending stresses they have to be transformed from polar coordinates to Cartesian according to [44]:

[1] Material taken from Leissa and Narita [43]
[2] Assumed to be the same as for Graphite/Epoxy
[3] Material taken from Suemasu, Kerth [42]

$$\begin{aligned}
\sigma_{xx} &= \frac{1}{2}\left(\sigma_{rr}+\sigma_{\theta\theta}\right)+\frac{1}{2}\left(\sigma_{rr}+\sigma_{\theta\theta}\right)\frac{x^2-y^2}{x^2+y^2} \\
\sigma_{yy} &= \frac{1}{2}\left(\sigma_{rr}+\sigma_{\theta\theta}\right)-\frac{1}{2}\left(\sigma_{rr}+\sigma_{\theta\theta}\right)\frac{x^2-y^2}{x^2+y^2} \\
\sigma_{zz} &= \sigma_{zz} \\
\sigma_{xy} &= \left(\sigma_{rr}-\sigma_{\theta\theta}\right)\frac{xy}{x^2+y^2} \\
\sigma_{xz} &= \sigma_{rz}\frac{x}{\sqrt{x^2+y^2}} \\
\sigma_{yz} &= \sigma_{rz}\frac{y}{\sqrt{x^2+y^2}}
\end{aligned} \tag{3.5.26}$$

Since we are dealing with a beam-like structure the values are only dependent on the x variable, simplifying the above to the following:

$$\begin{aligned}
&\sigma_{xx}=\sigma_{rr};\ \sigma_{yy}=\sigma_{\theta\theta};\ \sigma_{zz}=\sigma_{zz} \\
&\sigma_{xy}=0;\ \sigma_{xz}=\sigma_{rz};\ \sigma_{yz}=0
\end{aligned} \tag{3.5.27}$$

Now the two approaches can be combined into a single globally defined stress field. The next section discusses how this field is used to determine a possible delamination initiation site.

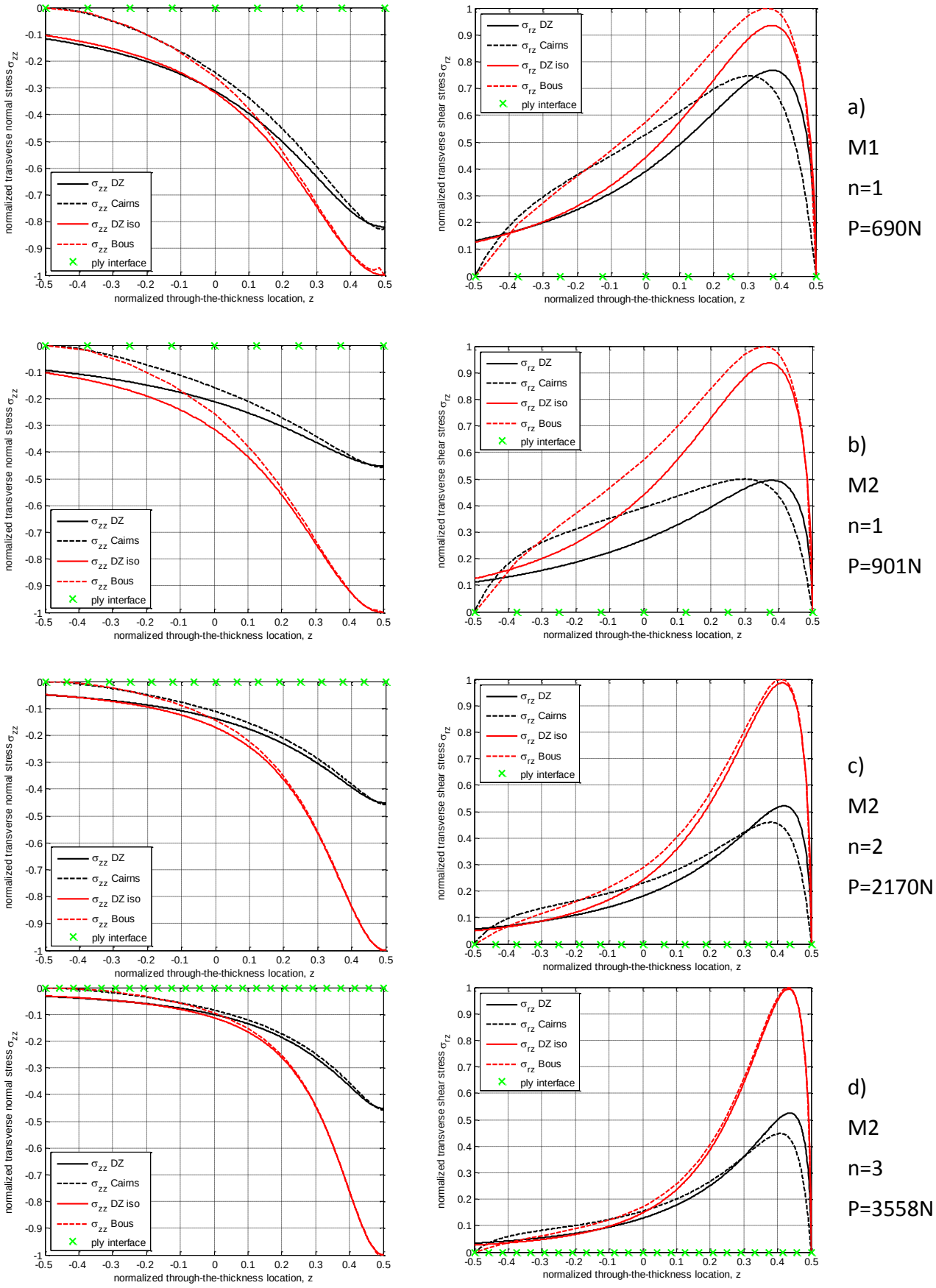


Figure 3.8 – Comparison of the two contact stress determination approaches for a $[0/45/90/-45]_{ns}$ laminate.

3.6 Failure analysis

Before a delamination can be analysed in detail we need to establish at which interface the first and any subsequent delaminations will occur and, equally important, at which load level. To this end a vast range of failure criteria have been proposed which predict the onset of a delamination (see [5]). All of these criteria are based in one way or the other on one or more three dimensional stress components. The selected criteria for this project will briefly be discussed alongside with the main assumptions regarding failure initiation.

3.6.1 Assumptions

At this point a major assumption will be made with respect to the normal occurring failure modes. As mentioned before delaminations are always accompanied and preceded by other failure modes unless introduced artificially into a laminate. These preceding failure modes are localized in their nature and have a lesser to negligible influence on the compression after impact (CAI) performances of laminates [3]. The inclusion of these failure modes in the analytical model would certainly increase the complexity of it which would work counterproductive to the use this model should be put to. Keeping all of that in mind, and based on the fact that delamination initiation criteria are present, it was decided to focus solely on delaminations as the only failure mode present in the model. In all of the following considerations we will work with symmetry conditions, i.e. we assume that delaminations develop symmetrically with respect to the line of impact. In this context it is important to question the minimum size of a delamination from a macromechanics point of view. It has been found that a crack can be considered a delamination if it exhibits sufficient influence on the structural behaviour of the plate. The minimum length for that has been found at delamination lengths of about 100 times the diameter of a single fibre which corresponds to approximately 4-5 times the ply thickness. It is also commonly known that delaminations occur almost exclusively between layers of different orientation. Alternatively, some researchers say that multiple layers of the same orientation act as one single layer, which is why it is very difficult for a delamination to form within that layer. Based on this aspect Liu [35] formulated his criterion for the prediction of delamination area taking into account the bending stiffness mismatch between two adjacent layers. Therefore, in this work we shall only consider delaminations between differently orientated layers.

3.6.2 Failure criterion selection

When looking at the delamination initiation criteria reviewed by Orifici, Herszberg [5] it becomes clear that most of them have been developed for in-plane loading, since the transverse normal stress σ_{zz} is always compared to the transverse normal tensile strength. However, the nature of the loading considered here clearly leads to compressive transverse normal stresses. One criterion that differs between tensile and compressive transverse normal stresses is the one proposed by Brewer and Lagace [20], although developed for an in-plane loaded specimen. Another one not listed by [5] has been developed by Hou, Petrinic [21] and this considers the sign of the transverse normal stress by adjusting the criterion for different cases. The criterion from Brewer and Lagace [20] takes the following form:

$$\left(\frac{\sigma_4}{S_4}\right)^2+\left(\frac{\sigma_5}{S_5}\right)^2+\left(\frac{\sigma_3^-}{Z_c}\right)^2\geq 1 \qquad (3.6.1)$$

where S_4, S_5 and Z_c are the transverse shear and compressive strengths. Note that the tensile transverse normal stress has been left out since in the area of interest only compressive stresses occur. The criterion proposed by Hou, Petrinic [21] can be summarized as follows:

$$
\begin{aligned}
e_I^2 &= \left(\frac{\sigma_3}{Z_t}\right)^2 + \frac{\sigma_4^2+\sigma_5^2}{S_4^2\left(d_{ms}d_{fs}+\delta\right)} \geq 1 \text{ for } \sigma_3 \geq 0 \\
e_I^2 &= \frac{\sigma_4^2+\sigma_5^2-8\sigma_3^2}{S_4^2\left(d_{ms}d_{fs}+\delta\right)} \geq 1 \text{ for } -\sqrt{\frac{\left(\sigma_4^2+\sigma_5^2\right)}{8}} \leq \sigma_3 < 0 \\
e_I^2 &= 0 \text{ for } \sigma_3 < -\sqrt{\frac{\left(\sigma_4^2+\sigma_5^2\right)}{8}}
\end{aligned}
\tag{3.6.2}
$$

The complicating issue with this criterion is that the coefficients d_{ms}, d_{fs} and δ are based on the matrix and fibre failure present in the laminate. While this certainly helps to obtain a more accurate picture it relies on FE analyses to obtain this failure state preceding the delamination.

Frequent occurrences during impact are delaminations that form close to the back face due to the high in-plane tension loads. The criterion shown in (3.6.1) does not take this component into account. However, there are a number of criteria, which include the in-plane normal component in the same quadratic fashion as any of the other stress components, which would result in an adjusted criterion based on Brewer and Lagace [20]:

$$
\left(\frac{\sigma_4}{S_4}\right)^2 + \left(\frac{\sigma_5}{S_5}\right)^2 + \left(\frac{\sigma_3^-}{Z_c}\right)^2 + \left(\frac{\sigma_1}{X_t}\right)^2 \geq 1 \tag{3.6.3}
$$

In all of the above shown criteria the stress values are based on the individual ply coordinate system. From the two stress analyses carried out in sections 3.4 and 3.5 only the global values are known. In order to be able to apply the failure criteria we need to transform the globally defined stress components to those defined in ply coordinates by the well-known expression:

$$
\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 0 & 0 & 0 & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & 0 & 0 & 0 & -2\sin\theta\cos\theta \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 & \sin\theta & \cos\theta & 0 \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & 0 & 0 & 0 & \cos^2\theta-\sin^2\theta \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xz} \\ \sigma_{yz} \\ \sigma_{xy} \end{Bmatrix} \tag{3.6.4}
$$

Due to the possibility of directly applying (3.6.3) as a failure criterion it will be used as the current tool for deciding whether or not an interface is failing and a delamination is initiated.

3.7 Delaminated beam analysis

In this section the analysis for a beam with multiple delaminations embedded in its structure is presented. The derivations are based on LBT and feature two different approaches with respect to the contact formulation between the beam segments, constrained and unconstrained. The results of this section will be used in section 3.8 to compute the energy release rate (ERR) and stress intensity factors (SIF) and in section 3.9 for the computation of the equivalent bending compliance.

3.7.1 Assumptions

The main assumptions are made when formulating the contact between the delaminated segments. Two different cases are being distinguished, namely constrained and unconstrained contact. During constraint contact it is assumed that the delaminated beam segments move together and do not separate. This constraint is relaxed when assuming unconstrained contact where individual beam segments are free to move with respect to other segments. However, while an opening of two segments is considered physically allowable this formulation also includes the physically inadmissible interpenetration of beam segments. In terms of the ERR choosing one formulation over the other this has important implications. The fact that no opening is allowed in combination with constrained contact means that no mode I delamination can occur and the computed ERR can be compared directly with the critical ERR during a pure mode II delamination. The unconstrained contact is able to predict opening of beam segments, indicating a mode II delamination. The advantage of using either of these approximations lies in the fact that they principally allow for closed form analytical expressions for the displacement functions of the beam segments, certainly when simple cases are concerned. A more advanced formulation of the contact between two beam segments models the interface as a series of springs, which are only active when these segments are in contact. While this method is the most accurate of the three contact formulations [34]it is also the most complex one. It involves numerical solution tools, as the definition of the spring is defined on the relative displacement of two or more beam segments, which is not known a priori. Since such a modelling is outside the scope of this thesis this form of contact will not be considered further.

As already mentioned the derivations shown are based on LBT, i.e. that all assumptions pertaining to this particular theory are valid here as well (see section 3.4). However, we shall relax the assumptions regarding symmetry and balance of the layup. Since delaminations may occur randomly the resulting sublaminates are most likely neither symmetric nor balanced. In these cases the reduced bending stiffness matrix will be computed which takes the presence of the coupling matrix due to asymmetry in the layup into account.

In all the derivations we still assume small deflections, i.e. that no in-plane forces will develop due to membrane effects. Also no shear forces will develop between the layers, i.e. any contact between delaminated beam segments is considered frictionless and the beam segments are continued to be non-shear deformable.

3.7.2 Analytical approach

The situation considered is shown in Figure 3.9. The pressure distribution shown in Figure 3.9 b) can be written as:

$$p_{k,k+1}(x) = k_{k,k+1}(x)\left(w_k(x) - w_{k+1}(x)\right) \tag{3.7.1}$$

The spring stiffness $k_{k,k+1}$ is based on the relative displacement of the two segments and represents the stiffness of the interface of two segments in contact. In the two extreme cases that are being considered here this pressure vanishes. For constrained contact we assume that $w_k = w_{k+1}$ while in unconstrained contact the stiffness of the interface is simply 0. Both result in $p_{k,k+1} = 0$. With that in mind the equilibrium for a beam segment as shown in Figure 3.9 b) can be written as:

$$\begin{aligned} M_k' - V_k &= 0 \\ V_k' &= 0 \end{aligned} \tag{3.7.2}$$

where the prime denotes a derivative with respect to the coordinate x. Since shear deformations are not included the relation between curvature and out-of-plane deformation can be written as:

$$\kappa_k = -w_k'' \tag{3.7.3}$$

The constitutive relation for a beam segment k relating the bending moment to the curvature is given as:

$$\begin{aligned} M_k &= (EI)_k \, \kappa_k = -(EI)_k \, w_k'' \\ &\text{with} \\ (EI)_k &= E_{xx}^b I_{yy} \end{aligned} \tag{3.7.4}$$

where E_{xx}^b is given in equation (3.3.12). Finally, combining equations (3.7.2) to (3.7.4) yields the governing differential equation for the beam segment k:

$$(EI)_k \, w_k^{IV} = 0 \tag{3.7.5}$$

The general solution for each beam segment is obtained by integrating the last equation four times resulting in:

$$w_k(x) = C_{k,1} + C_{k,2} x + C_{k,3} x^2 + C_{k,4} x^3 \tag{3.7.6}$$

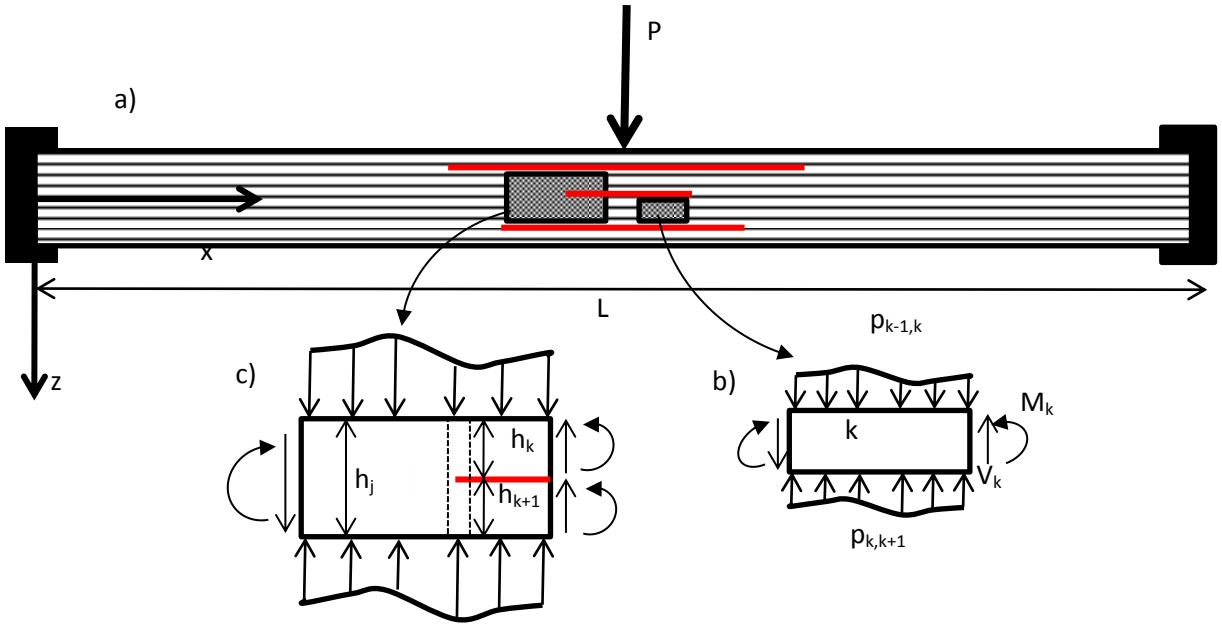


Figure 3.9 - delaminated composite beam under point load. a) global overview, b) isolated beam element k with forces acting on it, c) detail of split beam segment

While the definition of E_{xx}^{b} for the undamaged part 0 poses no problem, some measures need to be taken when defining E_{xx}^{b} for any of the delaminated segments. As already mentioned, the created sublaminates probably do not satisfy the conditions of balance and symmetry with respect to the layup anymore. This means that the B-matrix, which governs the coupling between in- and out-of-plane behaviour, is now fully populated and not zero anymore [4]. In order to take these effects into account but still be able to work with simplified expressions the reduced form of the D-matrix can be computed according to:

$$D_{red} = D - BA^{-1}B \tag{3.7.7}$$

The inverse of the D_{red} matrix then yields the value needed in order to compute E_{xx}^{b} for any of the delaminated segments.

3.7.2.1 Unconstrained contact

In the case of unconstrained contact every beam segment needs to be treated separately, for a relatively simple case of a singly delaminated beam this already leads to 12 equations that need to be solved simultaneously. In order to explain the implementation of this particular contact formulation consider the following case shown in Figure 3.10. It depicts half a beam, either clamped or simply supported, it also shows the numbering scheme that has been adopted here. The beam segments are numbered from top to bottom for every delamination tip. Reflecting this numbering scheme is the interface matrix. It helps connecting the right elements with each other when it comes to formulating boundary and continuity conditions. The interface matrix for the current example looks like:

$$\text{interface} = \begin{bmatrix} 1 & 3 & 3 & 3 \\ 1 & 4 & 4 & 4 \\ 2 & 2 & 5 & 7 \\ 2 & 2 & 5 & 8 \\ 2 & 2 & 6 & 6 \end{bmatrix} \tag{3.7.8}$$

The number of rows reflects the number of delaminated segments at mid-span while the number of columns represents the number of different delamination length. Every delamination is clearly identified by a_i and z_i, its length and position within the beam, respectively. The boundary conditions for this beam are given as:

$$\left.\begin{array}{c} w_k = w_{k+1} \\ w_k' = 0 \\ \dfrac{P}{2} = \displaystyle\sum_k P_k \end{array}\right\} \text{at } x = \frac{L}{2}$$

$$\left.\begin{array}{c} \text{clamped:} \\ w_0 = w_0' = 0 \Leftrightarrow C_{0,1} = C_{0,2} = 0 \\ \text{simply supported:} \\ w_0 = M_0 = 0 \Leftrightarrow C_{0,1} = C_{0,3} = 0 \end{array}\right\} \text{at } x = 0 \tag{3.7.9}$$

Here, k refers to the elements of the last column of the interface matrix in (3.7.8). Next to the boundary conditions there are kinetic and static continuity conditions at a delamination tip. If for example we take crack tip x_2 and look at it as shown in Figure 3.9 c) we can write the continuity equation as:

$$\left.\begin{aligned} w_1(x_2) &= w_3(x_2) = w_4(x_2) \\ w_1'(x_2) &= w_3'(x_2) = w_4'(x_2) \\ M_1(x_2) &= \sum_n M_n(x_2), \text{ for n} = 3,4 \\ P_1(x_2) &= \sum_n P_n(x_2), \text{ for n} = 3,4 \end{aligned}\right\} \text{ at } x = x_2 = \frac{L}{2} - a_2 \qquad (3.7.10)$$

The interface matrix previously defined helps in formulating these conditions properly and is further used to determine the ERR and SIF as will be shown later. Imposing both boundary and

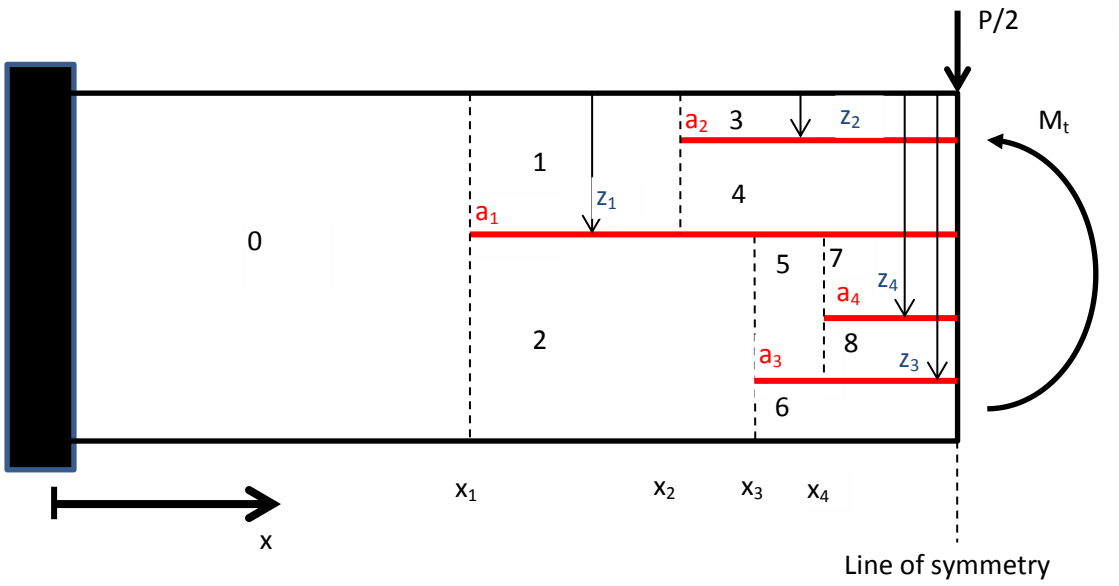


Figure 3.10 - randomly delaminated half structure for unconstrained contact

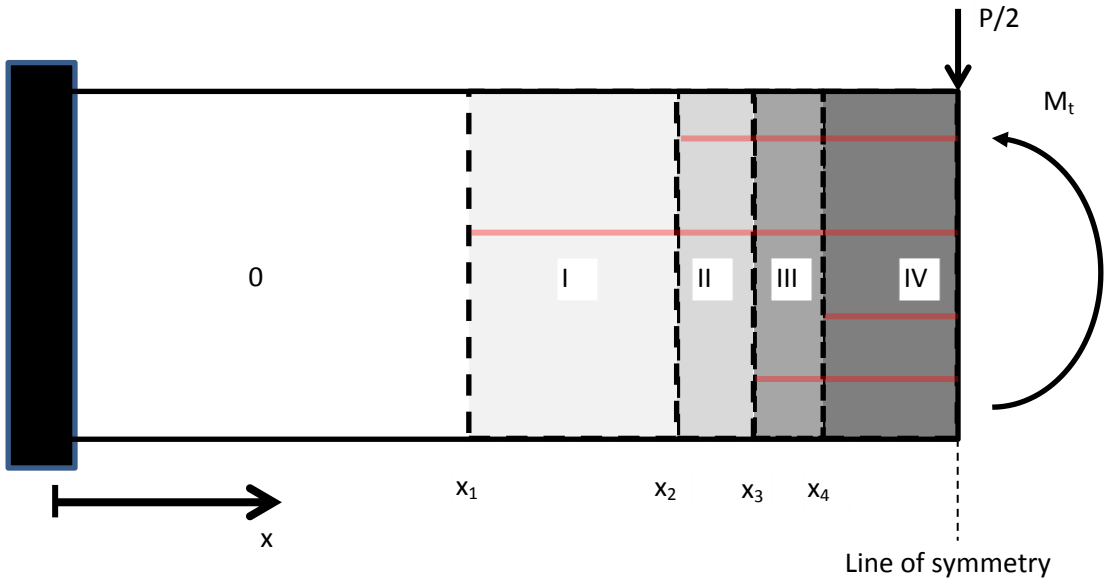


Figure 3.11 - Randomly delaminated half structure for constrained contact

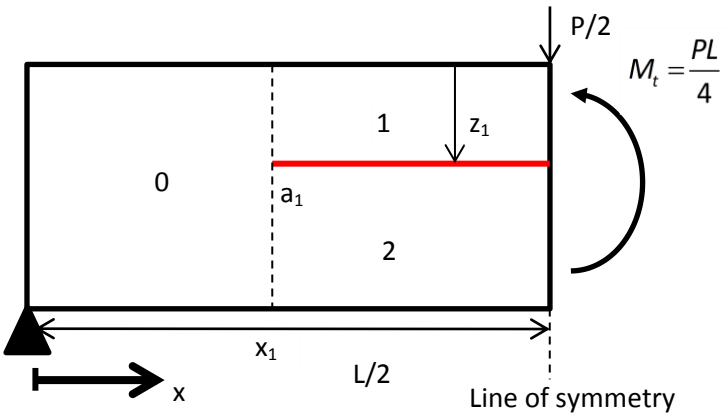


Figure 3.12 – Singly delaminated structure used as example

continuity conditions for every delamination tip will result in a system of equations that needs to be solved simultaneously for the unknown displacement coefficients $C_{k,I}$ – $C_{k,4}$, for k = 0..7 in this example.

3.7.2.2 Constrained contact

We consider again the case depicted in Figure 3.10. This time, however, we approach it using the constrained contact formulation. This effectively changes the situation to the one shown in Figure 3.11, where the individual beam segments have been merged to sections bounded by the different lengths of the delaminations. Each section now corresponds to a column of the interface matrix, i.e. that the interface matrix immediately shows which segments make up the section.

In deriving the equations for the displacement we still use equation (3.7.6) as starting point. We can write the internal moment in each of the sections as:

$$M(x) = -(EI)_k w_k''(x), \text{ for } k = 0..IV \tag{3.7.11}$$

Note that k now uses Roman numerals as shown in Figure 3.11. For the case of a simply supported beam M is given as $0.5Px$. This means that two of the four unknowns in equation (3.7.6) are already taken care of, namely $C_{k,3}$ (=0) and $C_{k,4}$ (=-P/2). Double integration of equation (3.7.11) leads to:

$$\begin{aligned} w_k' &= -\frac{Px^2}{4(EI)_k} + C_{k,2} \\ w_k &= -\frac{Px^3}{12(EI)_k} + C_{k,2}x + C_{k,1} \end{aligned} \tag{3.7.12}$$

It has to be noted that the flexural rigidity is now defined for each section according to:

$$(EI)_i = \sum_k (EI)_k \tag{3.7.13}$$

where the summation is carried out over the unique entries in the i^{th} column of the interface matrix. Just like in the case of the unconstrained contact we can formulate boundary and continuity equations. For the case of a simply supported beam these can be formulated as:

$$\left.\begin{aligned} w_0 = 0 \Leftrightarrow C_{0,1} = 0 \\ w_{end}' = 0 \\ w_i(x_i) = w_{i+1}(x_i) \\ w_i'(x_i) = w_i'(x_i) \end{aligned}\right| \begin{aligned} &\text{at } x = 0 \\ &\text{at } x = L/2 \\ &\text{at } x = x_i = L/2 - a_i \end{aligned} \tag{3.7.14}$$

In the case of a single length delaminated beam, simply supported at its end (see Figure 3.12), this set of equations allows for a closed form solution of the entire displacement field. When starting with the second expression in (3.7.14) the coefficient $C_{1,2}$ from equation (3.7.12) is readily available as:

$$C_{1,2} = \frac{PL^2}{16(EI)_1} \tag{3.7.15}$$

By observation it is always possible to find the next equation from the set of expressions in (3.7.14) which only contains one unknown at a time. Carrying on with the example of single length delaminated beam the next equation would be the rotation continuity at x_1:

$$\begin{gathered} w_0' = w_1' \\ -\frac{Px_1^2}{4(EI)_0} + C_{0,2} = -\frac{Px_1^2}{4(EI)_1} + C_{1,2} \\ C_{0,2} = \frac{Px_1^2}{4}\left(\frac{1}{(EI)_0} - \frac{1}{(EI)_1}\right) + \frac{PL^2}{16(EI)_1} \end{gathered} \tag{3.7.16}$$

Finally, the last unknown coefficient $C_{1,1}$ can be solved for by requiring displacement continuity at the interface:

$$\begin{gathered} w_0 = w_1 \\ -\frac{Px_1^3}{12(EI)_0} + C_{0,2}x_1 = -\frac{Px_1^3}{12(EI)_1} + C_{1,2}x_1 + C_{1,1} \\ C_{1,1} = -\frac{Px_1^3}{12}\left(\frac{1}{(EI)_0} - \frac{1}{(EI)_1}\right) + x_1\left(C_{0,2} - C_{1,2}\right) \\ C_{1,1} = -\frac{Px_1^3}{12}\left(\frac{1}{(EI)_0} - \frac{1}{(EI)_1}\right) + \frac{Px_1^3}{4}\left(\frac{1}{(EI)_0} - \frac{1}{(EI)_1}\right) \\ C_{1,1} = \frac{Px_1^3}{6}\left(\frac{1}{(EI)_0} - \frac{1}{(EI)_1}\right) \end{gathered} \tag{3.7.17}$$

In case of multiple different length delaminations this process will have to be repeated multiple times in order to obtain the full displacement filed. When the beam is clamped at both ends the same procedure has to be followed starting from equation (3.7.11). However, now the moment includes the previously unknown root moment which has to be incorporated as an unknown to be solved for. Therefore, the solution will not be as simple to obtain as it is for the case of a simply supported beam.

3.8 Linear elastic fracture mechanics approach

The growth of delaminations and effects related to crack forming is generally studied within the field of Linear Elastic Fracture Mechanics (LEFM). As already mentioned before, there are two distinct approaches within this field: the global view that uses energy balance in order to compute the so-called ERR and a more local view that looks at the near crack-tip stress field and computes the so-called Stress Intensity Factor (SIF). Both approaches are valid and have been used frequently over the past years. In this section both approaches will be discussed. The concept of the ERR has commonly been used to determine the development of one or several delaminations and will be used here in the same way. In order to get an update on the stress state of a delaminated stack it is important to know what the local stress field around this tip looks like, for which the concept of the SIF normally is used.

3.8.1 Assumptions

The main assumptions made in this section are listed below.

- The use of LEFM already implicitly assumes the material under consideration to behave in a linearly elastic fashion.
- Only half of the structure will be examined in the following, implying that symmetry conditions are applied with respect to the z-axis. This assumes that the delaminations expand symmetrically in both directions, i.e. in case of a beam in a self-similar way.
- Non-shear deformable beams will be considered.
- For the computation of the SIF it is assumed that any delamination tip can be extracted as shown in Figure 3.9 c) and that in this case the top and bottom surfaces are traction free.
- When using the solution proposed by Suo and Hutchinson [1] it is assumed that the interactivity between several delamination tips is accounted for in the magnitude of the SIF. The resulting singular stress fields will be superimposed.

3.8.2 Energy release rate computation

The solution from section 3.7 will be used in this section to compute the ERR for the beam in question. However, the full solution of the system of equations is not always necessary as we will see for certain cases. It remains important to know the displacement of the beam, certainly underneath the impactor, as this information will be used in section 3.9.

We have mentioned earlier that the concept of the ERR is related to the global energy approach within the field of LEFM. This justifies the effort that has been put into determining the displacement field of the entire beam and implies that the result is a single value combining all modes. While for the constrained contact formulation this is not an issue, only mode II delamination can occur, it is erroneous for the unconstrained contact formulation, where the opening mode can occur as well. Both contact formulations are considered extremes in terms of representing the actual behaviour which is why they both should be regarded with care [28].

The method of obtaining the ERR will be explained next, followed by a brief discussion about the differences in the two contact formulation for different cases.

3.8.2.1 Derivation of the energy release rate

In its most general form the ERR following the applied displacement approach can be written as:

$$G = -\frac{\partial U}{\partial A} \tag{3.8.1}$$

where U is the strain energy of the body and A the area created during the formation of a crack. This formulation is based on the work of Griffith and has been used by many researchers in the past [45]. One has to note, that when using the applied displacement approach the derivative of U with respect to A contains a derivative of the load acting on the structure P with respect to A via the chain rule, which is always negative [22]. In absence of any in--plane forces we can write the strain energy for a beam-like structure as:

$$U = \frac{1}{2}\int_{Vol} \sigma_{xx}\varepsilon_{xx} dV = \frac{1}{2}\int_0^L E_{xx}^b I_{yy} \left(\frac{\partial^2 w}{\partial x^2}\right)^2 dx \tag{3.8.2}$$

This expression is valid only for an undamaged beam. The presence of one or several delaminations leads to a division of the entire length into several pieces. For a generally delaminated beam such as shown in Figure 3.10 the strain energy can be written as:

$$\begin{aligned} U &= \frac{1}{2}\left\{ \int_0^{x_1} (EI)_0 \left(\frac{\partial^2 w_0}{\partial x^2} \right)^2 dx + \sum_j \sum_i \int_{x_j}^{x_{j+1}} (EI)_i \left(\frac{\partial^2 w_i}{\partial x^2} \right)^2 dx \right\} \\ &= \frac{1}{2}\left\{ \int_0^{x_1} (EI)_0 \left(2C_{0,3} + 6C_{0,4}x \right)^2 dx + \sum_j \sum_i \int_{x_j}^{x_{j+1}} (EI)_i \left(2C_{i,3} + 6C_{i,4}x \right)^2 dx \right\} \\ &= \frac{1}{2}\left\{ \begin{array}{l} (EI)_0 \left[4C_{0,3}^2 x_1 + 12C_{0,3}C_{0,4}x_1^2 + 12C_{0,4}^2 x_1^3 \right] + \\ \sum_j \sum_i (EI)_i \left[4C_{i,3}^2 \left(x_{j+1} - x_j \right) + 12C_{i,3}C_{i,4} \left(x_{j+1} - x_j \right)^2 + 12C_{i,4}^2 \left(x_{j+1} - x_j \right)^3 \right] \end{array} \right\} \end{aligned} \tag{3.8.3}$$

Here j represents the column of the interface matrix and the summation in i is over the unique elements of that column. The last item of the integration bound vector is L/2. The change in area A that is created during the forming of a delamination can be written as:

$$\partial A = b \partial a_j \tag{3.8.4}$$

where a_j indicates the considered delamination length. This means that now we only have to compute the derivative of U with respect to the delamination length a_j. It will be shown later that for either the constrained contact or the simple case of one delamination this differentiation can be carried out analytically. The more general case will make use of a numerical forward difference scheme, which has been tuned such that for the simple cases this solution corresponds well with the analytical ones. When dealing with multiple delaminations, it is often assumed that all delaminations are equally spread throughout the thickness of the laminate and that they all growth in a simultaneous way. In this work, however, when delaminations of the same length occur the following approach will be adopted. Subsequently every delamination will be assessed individually by only changing its length by an amount Δa_j. The differentiation of the strain energy wrt this changing length will then be computed based on the forward difference scheme until the change Δa_j approaches 0. By assigning each delamination tip its own value for the ERR one allows for both length and relative placement within the lamination stack to influence the growth behaviour of all the delaminations present (Suemasu and Majima [29] and Andrews, Massabó [34]). These effects will be shown later. In contrast to most of the published literature the stacking information should be preserved as long as possible rather than homogenizing the material and spreading the delaminations equally. This should ensure potential suitability for optimization routines to be applied.

When considering the constrained contact formulation (see Figure 3.11) the expression shown in (3.8.3) simplifies to:

$$U = \frac{1}{2} \sum_l \int_{x_l}^{x_{l+1}} (EI)_l \left(\frac{\partial^2 w_l}{\partial x^2} \right)^2 dx \tag{3.8.5}$$

Using equation (3.7.11) the strain energy can be computed with the knowledge of only the internal bending moment distribution. In the case of a clamped beam this distribution depends on the root moment, which is obtained by solving the system of linear equations. In the case of a simply supported beam, the internal bending moment distribution is simply Px/2. This allows us to write the strain energy for half the beam structure as:

$$U = \frac{P^2}{24}\sum_I \frac{\left(x_{I+1}^3 - x_I^3\right)}{\left(EI\right)_I},\ I = 0..n_{sec} \tag{3.8.6}$$

where, n_{sec} is the number of sections as defined in Figure 3.11. Note that $x_0 = 0$ and $x_{nsec+1} = L/2$. Differentiating this expression with respect to the considered delamination length will once more yield the ERR:

$$G_I = -\frac{\partial U}{\partial A} = \frac{P^2}{8n_d b}\left(\frac{L}{2} - a_{I+1}\right)^2\left[\frac{1}{\left(EI\right)_{I+1}} - \frac{1}{\left(EI\right)_I}\right] \tag{3.8.7}$$

This shows that for the case of a simply supported beam the ERR can be computed by only knowing the geometry of all delaminations present. The only problem with this particular equation is that it does compute the same value for equal length delaminations in contrast to the approach outlined above.

3.8.2.2 Example: two unequal and randomly distributed delaminations

Trying to visualize all that has been said above for a general case of a delaminated beam is impossible. Rather we shall try to bring more insight to this matter by looking at a two-delamination system as shown in Figure 3.13. Similar studies have been carried out by Andrews for the case of a cantilevered beam [28]. We want to investigate the influence the longer delamination has on the lower based on the relative position of the two. To this end we will compute the ratio of ERR_I/ERR_{I0}. Here ERR_{I0} is the ERR of the lower delamination without the presence of the upper one. The result of this computation can be mapped and is shown for this case in Figure 3.14. The position of the two delaminations in terms of the interface they are located at is given on the ordinates. Note, that the lower delamination is measured with respect to the back face. The straight diagonal line represents the limit of the map, as combinations of delamination positions beyond this line are not possible. The two areas separated by the discontinuous line represent the two effects that can occur in multiply delaminated structures. To the left of this line the ratio computed is larger than one, which means that in this area the growth of the lower delamination is amplified by that of the upper one. In the area between the two lines the ratio is less than one, i.e. we have the reverse effect: the ERR of the lower delamination is shielded by that of the upper one. A qualitatively similar map can be created if the lengths would be swapped.

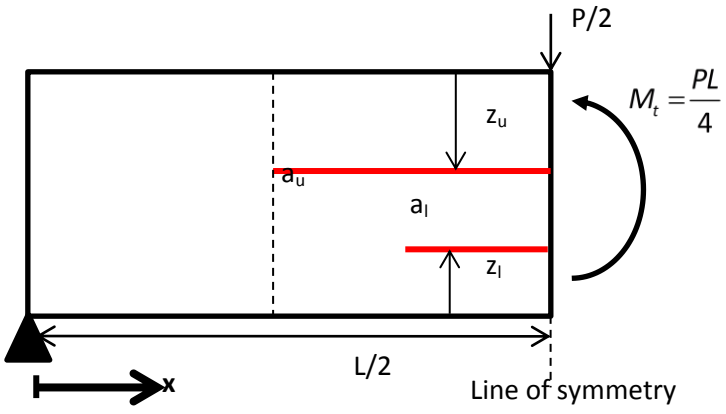


Figure 3.13 - Two delamination system

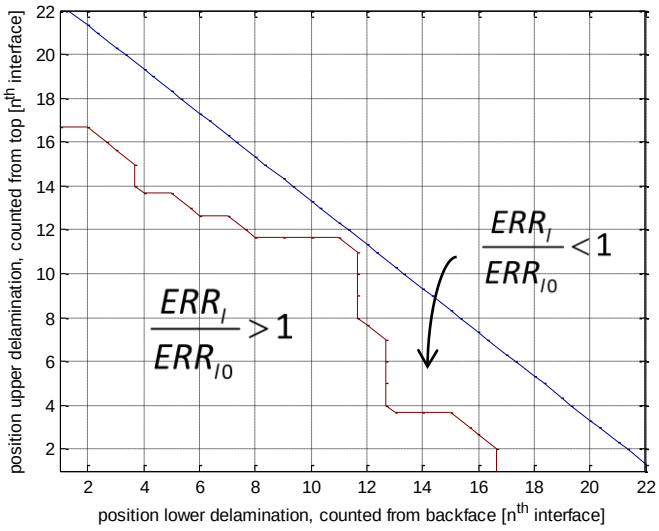


Figure 3.14 - Map of amplification and shielding of the lower delamination in presence of a longer upper delamination

If we pick a location from each of the two areas we can draw the ERR of the lower delamination for a changing delamination length and a fixed upper delamination length. This has been done in Figure 3.15. In Figure 3.15 a) we can clearly see the amplification effect mentioned earlier on the lower delamination. As the length of the lower delamination approaches and surpasses that of the upper one this effect diminishes. The shielding effect can be seen in Figure 3.15 b). While for the maps drawn in Figure 3.14 there was no significant difference between the constrained and unconstrained contact, a clear difference can be seen in Figure 3.15. This shows that both length and relative position of the delaminations within a structure play a role in the interaction between the delaminations. In both figures the discontinuity at the point where the lower delamination reaches the length of the upper delamination can be observed. This discontinuity can only be resolved using more advanced methods.

At this point one last thing should be mentioned. Some of these expressions allow for the delamination length to go to zero, implying that it might be possible to compute the ERR for the initiation of a delamination. However, as these expressions have been derived within the context of LEFM this is strictly speaking not correct, as LEFM always assumes a pre-existing crack. In order to initiate cracks or delaminations we will therefore refer to stress-based failure criteria as discussed above. In presence of delaminations SIFs will influence the stress state. The derivation of these will be discussed in the following section.

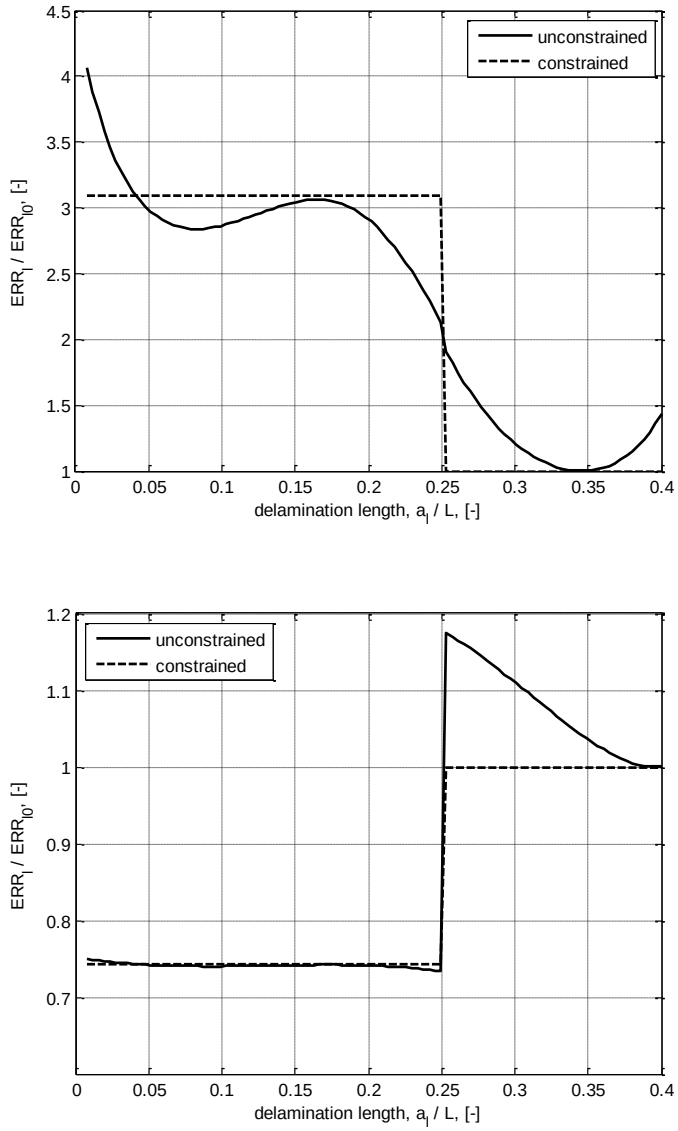


Figure 3.15 - Development of ERR of lower delamination for changing delamination length. a) Example for amplification, b) example for shielding

3.8.3 Stress intensity factors at interface cracks

Being able to compute the ERR for all delaminations within a structure is only part of the problem that has to be solved. In order to determine the next delamination site one has to be able to predict the stress state in presence of an existing crack. Naturally, this problem has received a lot of attention in the case of isotropic material, where cracks are placed randomly throughout the material and where those can grow in a direction determined by the loading. Here we are dealing with a particular type of crack, a delamination, which only occurs at interfaces within a laminated structure. The main difference between cracks in homogenous and dissimilar cracks lies in the fact that in the latter case delaminations occur in a mixed-mode fashion [22]. Another problem with interface cracks that is linked to this mixed-mode effect concerns the stresses near the crack-tip. It has been shown that these oscillate very strongly if the point of interest approaches the crack-tip.

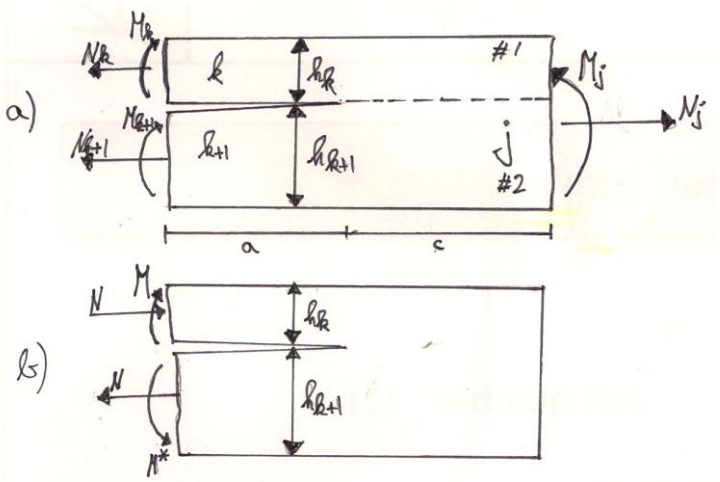


Figure 3.16 - Geometry of generally applied edge forces at delamination tip (Figure taken from [1])

This makes the accurate prediction of the individual modes difficult, however, the total ERR based on the interface SIF can always be defined clearly [25].

The analysis shown here is based on the work of Suo and Hutchinson [1]. They derived a solution for a semi-infinite crack lying at an interface between two infinite isotropic elastic layers under general loading. The situation is depicted in Figure 3.16. Basically, Figure 3.16 a) is comparable to Figure 3.9 c) and refers to the assumption made above. From the global beam analysis we can obtain the acting moments as depicted in Figure 3.16 a). Once these are known a system of self-equilibrating edge loads can be defined as shown in Figure 3.16 b):

$$N = -C_1 \frac{M_j}{h_k}$$
$$M = M_k - C_2 M_j \tag{3.8.8}$$

where the moments M_j and M_k are defined in equation (3.7.4):

$$M_i = -\frac{(EI)_i}{b}\left(2C_{i,3} + 6C_{i,4}x_a\right),\ i = k, j \tag{3.8.9}$$

Here the coefficients $C_{i,3}$ and $C_{i,4}$ are obtained from solving the system of equations resulting from a multiply delaminated beam and x_a is the coordinate of the delamination tip.

The two coefficients C_1 and C_2 in equation (3.8.8) are given as:

$$C_1 = \frac{\Sigma}{I_0}\left(\frac{1}{\eta} - \Delta + \frac{1}{2}\right)$$
$$C_2 = \frac{\Sigma}{12 I_0} \tag{3.8.10}$$

with:

$$\Sigma = \frac{c_2}{c_1} \text{ with } c_i = \frac{\kappa_i + 1}{\mu_i}$$
$$\eta = \frac{h_k}{h_{k+1}},\ \Delta = \frac{1 + 2\Sigma\eta + \Sigma\eta^2}{2\eta(1+\Sigma\eta)} \tag{3.8.11}$$
$$I_0 = \frac{1}{3}\left\{\Sigma\left[3\left(\Delta - \frac{1}{\eta}\right)^2 - 3\left(\Delta - \frac{1}{\eta}\right) + 1\right] + 3\frac{\Delta}{\eta}\left(\Delta - \frac{1}{\eta}\right) + \frac{1}{\eta^3}\right\}$$

which means that they solely depend on the properties of the bi-material and the relative positioning of the crack. In equation (3.8.11) $\kappa_i = 3 - 4\nu_i$ for plane strain or $\kappa_i = (3-\nu_i)/(1+\nu_i)$ for plane stress and μ_i is the shear modulus of the segment. The value for the segment's Young's modulus has already been given in (3.3.12) and further explained in section 3.7. The values for the shear modulus μ_i and the Posisson's ratio ν_i are obtained in a similar fashion from CLPT as:

$$\mu_i = \frac{12}{h_i^3 d_{33}}$$
$$\nu_i = -\frac{d_{12}}{d_{11}} \tag{3.8.12}$$

This assumes that both layers above and below the delamination can be approximated by an isotropic material. It has to be stressed at this point that is a very simplifying assumption, however, in this way it was possible to define an interface crack between two sublaminates rather than an crack in a homogenized orthotropic material (see [28]). The SIF can now by expressed in complex form as:

$$K = K_1 + iK_2 \tag{3.8.13}$$

where $i = \sqrt{-1}$. By using the relation between the ERR and the complex SIF and defining the ERR in terms of the self-equilibrating loads it is possible to write K as:

$$K = \left(\frac{N}{\sqrt{Ah_k}} - ie^{i\gamma}\frac{M}{\sqrt{Ih_k^3}}\right)\frac{p}{\sqrt{2}}h_k^{-i\varepsilon}e^{i\omega} \tag{3.8.14}$$

Where:

$$A = \frac{1}{1+\Sigma(4\eta + 6\eta^2 + 3\eta^3)},\ I = \frac{1}{12(1+\Sigma\eta^3)}$$
$$\sin\gamma = 6\Sigma\eta^2(1+\eta)\sqrt{AI}$$
$$p = \sqrt{\frac{1-\alpha}{1-\beta^2}} \tag{3.8.15}$$
$$\varepsilon = \frac{1}{2\pi}\ln\frac{1-\beta}{1+\beta}$$

ϵ is known as the bi-material constant. α and β are Dundur's parameter defined as:

$$\alpha, \beta = \frac{\Gamma(\kappa_2 \pm 1) - (\kappa_1 \pm 1)}{\Gamma(\kappa_2 + 1) + (\kappa_1 + 1)} \tag{3.8.16}$$

where $\Gamma = \mu_1 / \mu_2$. In equation (3.8.14) only ω is still unknown. This dimensionless parameter is a function of the parameters α, β and η. Suo and Hutchinson [1] state that this parameter has to be

solved once numerically for a certain loading case and is then applicable to other loading cases. Further the dependence of ω on the other three parameters is relatively weak. In this work it has been assumed that ω varies linearly only with respect to η:

$$\omega = 52.1° - 3\eta° \tag{3.8.17}$$

One can express the complex interface intensity factor in terms of the classical mode I and mode II SIF:

$$Kh_k^{i\varepsilon} = K_I + iK_{II} \tag{3.8.18}$$

The two different SIF modes can now be written as the real and complex part of equation (3.8.18):

$$\begin{aligned} K_I &= \Re\left(Kh_k^{i\varepsilon}\right) = \frac{p}{\sqrt{2}}\left[\frac{P}{\sqrt{Ah_k}}\cos\omega + \frac{M}{\sqrt{Ih_k^3}}\sin\omega + \gamma\right] \\ K_{II} &= \Im\left(Kh_k^{i\varepsilon}\right) = \frac{p}{\sqrt{2}}\left[\frac{P}{\sqrt{Ah_k}}\sin\omega - \frac{M}{\sqrt{Ih_k^3}}\cos\omega + \gamma\right] \end{aligned} \tag{3.8.19}$$

It has to be mentioned at this point that we assume that η ranges from 0 to 1 or that $h_k < h_{k+1}$. If that is not the case we have to redefine the equations adequately starting from (3.8.8). Note that all loads are defined per unit width.

The question arises now in how far this method is actually applicable to real-life structures. Suo and Hutchinson [1] only give qualitative guidelines. They say that the crack has to be long enough with respect to the layer k and that the crack should be placed well within the structure. In his thesis Andrews [28] is more precise. He calculates the uncertainties using advanced numerical tools and states that the application of his model, which is based on the work of Suo and Hutchinson, is accurate if the lengths a and c shown in Figure 3.16 are greater than a certain length $c_{\min} = h_i\lambda^{-1/4}, i = j,k,k+1$, where λ is the ratio of transverse to longitudinal Young's modulus. This ratio is smaller than 1 for fibre reinforced composites meaning that c_{min} is always greater than the largest height. We shall therefore always try to meet these conditions and apply the loads at the minimum required distance. If delamination tips approach each other, this will not be possible anymore. As no alternatives have been found with respect to computing the SIF in that particular case the same approach as outlined above for the determination of the ERR will be followed.

3.8.4 Near crack-tip stress field

The SIF obtained in equation (3.8.19) can be used to compute the singular stress field near a delamination tip. The derivation of this particular field is beyond the scope of this work. A comprehensive review of this matter is given in Sun and Jin [22]. It is commonly defined in polar coordinates (see Figure 3.17) and for a crack lying at an interface between two dissimilar bodies the following expressions apply:

$$\begin{aligned} \sigma_{xx,j} &= \frac{K_I}{2\sqrt{2\pi r}}\left[\omega_j f_{xx}^I - \frac{1}{\omega_j}\cos(\theta - \bar{\Theta})\right] - \frac{K_{II}}{2\sqrt{2\pi r}}\left[\omega_j f_{xx}^{II} + \frac{1}{\omega_j}\sin(\theta - \bar{\Theta})\right] \\ \sigma_{zz,j} &= \frac{K_I}{2\sqrt{2\pi r}}\left[\omega_j f_{zz}^I + \frac{1}{\omega_j}\cos(\theta - \bar{\Theta})\right] - \frac{K_{II}}{2\sqrt{2\pi r}}\left[\omega_j f_{zz}^{II} - \frac{1}{\omega_j}\sin(\theta - \bar{\Theta})\right], \text{ j = 1,2} \\ \sigma_{xz,j} &= \frac{K_I}{2\sqrt{2\pi r}}\left[\omega_j f_{xz}^I - \frac{1}{\omega_j}\sin(\theta - \bar{\Theta})\right] - \frac{K_{II}}{2\sqrt{2\pi r}}\left[\omega_j f_{xz}^{II} - \frac{1}{\omega_j}\cos(\theta - \bar{\Theta})\right] \end{aligned} \tag{3.8.20}$$

where:

$$\bar{\Theta} = \varepsilon \ln\left(\frac{r}{2a}\right) + \frac{\theta}{2}$$
$$\omega_1 = \exp\left[-\varepsilon(\pi - \theta)\right]$$
$$\omega_2 = \exp\left[\varepsilon(\pi + \theta)\right]$$
$$f_{xx}^{I} = 3\cos\bar{\Theta} + 2\varepsilon\sin\theta\cos(\theta + \bar{\Theta}) - \sin\theta\sin(\theta + \bar{\Theta})$$
$$f_{xx}^{II} = 3\sin\bar{\Theta} + 2\varepsilon\sin\theta\sin(\theta + \bar{\Theta}) + \sin\theta\cos(\theta + \bar{\Theta}) \qquad (3.8.21)$$
$$f_{zz}^{I} = \cos\bar{\Theta} - 2\varepsilon\sin\theta\cos(\theta + \bar{\Theta}) + \sin\theta\sin(\theta + \bar{\Theta})$$
$$f_{zz}^{II} = \sin\bar{\Theta} - 2\varepsilon\sin\theta\sin(\theta + \bar{\Theta}) - \sin\theta\cos(\theta + \bar{\Theta})$$
$$f_{xz}^{I} = \sin\bar{\Theta} + 2\varepsilon\sin\theta\sin(\theta + \bar{\Theta}) + \sin\theta\cos(\theta + \bar{\Theta})$$
$$f_{xz}^{II} = -\cos\bar{\Theta} - 2\varepsilon\sin\theta\cos(\theta + \bar{\Theta}) + \sin\theta\sin(\theta + \bar{\Theta})$$

The indices 1 and 2 denote the layer above and below the delamination, respectively. a is the length of the delamination and ϵ the bi-material constant defined in (3.8.15). For a homogenous material this constant reduces to zero and it can be shown that in this case the expressions in equation (3.8.20) reduce to the well-known classical singular stress field for the individual crack modes:

$$\sigma_{xx} = \frac{K_I}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right)$$
$$\sigma_{zz} = \frac{K_I}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \qquad (3.8.22)$$
$$\sigma_{xz} = \frac{K_I}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\sin\frac{\theta}{2}\cos\frac{3\theta}{2}$$

Here, the stress field for mode I is given. Respectively, for mode II the expressions are:

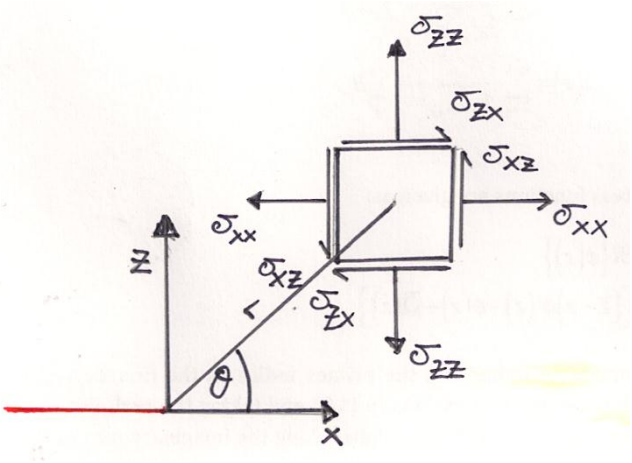


Figure 3.17 - coordinate system for near crack-tip stress field

$$\sigma_{xx} = -\frac{K_{II}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\left(2+\cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right)$$
$$\sigma_{zz} = \frac{K_{II}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \qquad (3.8.23)$$
$$\sigma_{xz} = \frac{K_{II}}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1-\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right)$$

The variables r and θ are shown in Figure 3.17. With these expressions the entire singular stress field around the delamination tip is defined when the tearing mode (see reference to crack modes) is neglected. Examples of these fields for a homogeneous material can be seen in Figure 3.18, where the singularities near the crack tip can clearly be observed. In this example $K_I = K_{II} = 1$. It can be seen from the expressions above that the order of magnitude of the SIF corresponds to the order of magnitude of the stress field surrounding the delamination tip. For a given stress component the corresponding mode is dominant. In the section these fields will be used to update the entire stress state in the damaged delaminated beam.

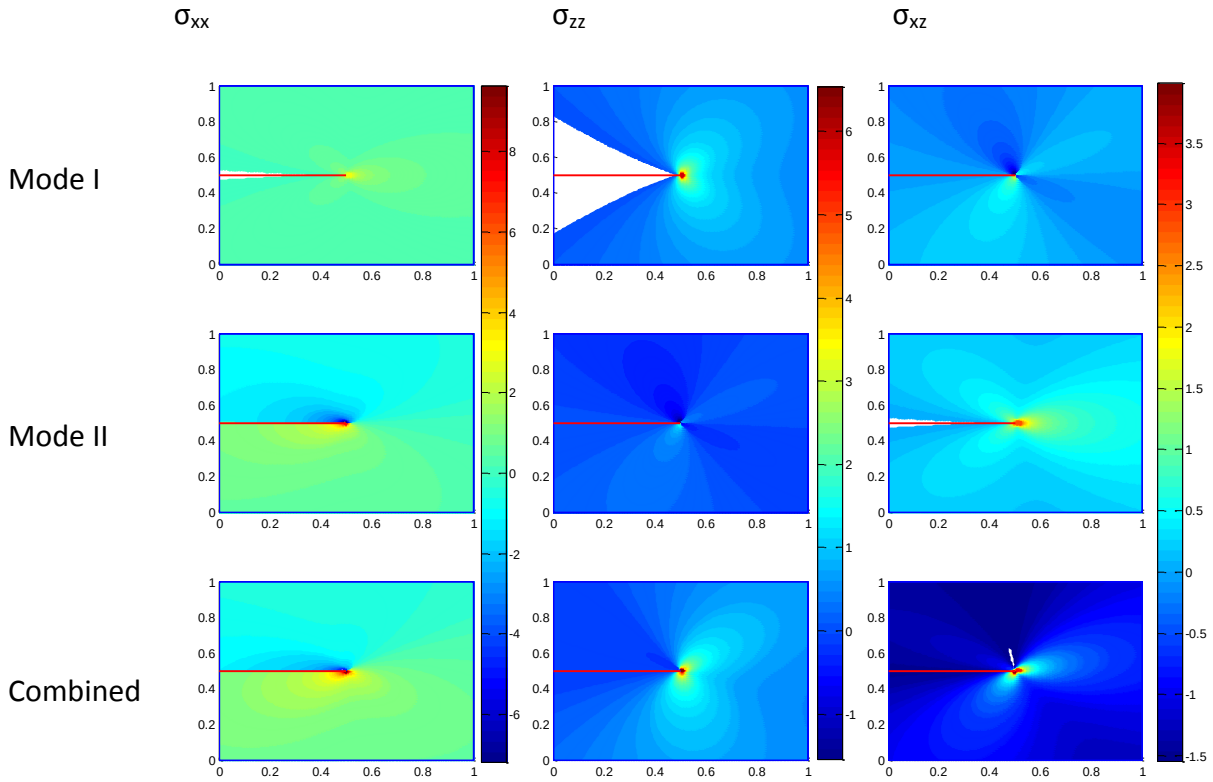


Figure 3.18 - Contour plots of singular stress field due to mode I and mode II opening (unit of the contours is Pascal)

3.9 Damage model

The final part of the analytical model is discussed in this section. After successfully detecting delamination initiation of one or several delaminations and computing both ERR and SIF for the current configuration, we will now discuss how these results will be used to update the applied force and determine growth of one or several delaminations.

3.9.1 Equivalent bending stiffness and applied load update

The occurrence of delaminations will alter the structural response of the beam, as it creates sublaminates of various lengths, thicknesses and elastic properties. This has a decreasing effect on the bending stiffness of

the beam and, because we use the applied displacement approach, consequently on the load acting on the beam.

The displacement distribution discussed in section 3.7 will be used in this section to determine the equivalent bending stiffness of and the new load acting on the beam. This equivalent bending stiffness serves as damage index and can be used to update the global structural analysis from section 3.4. It is based on the concept that the potential energy at the point of application is the same for the delaminated structure and a beam with equivalent bending stiffness. Let us denote $w_{c,u}$ as the central beam deflection obtained from section 3.7 due to a unit load and $w_{c,eq}$ as the same deflection for a composite beam with equivalent flexural rigidity $(EI)^*$ under unit load given by:

$$w_{c,eq} = \underbrace{\frac{L^3}{48C\left(EI\right)^*}}_{k_{b,eq}^{-1}} \text{ with } C = \begin{cases} 1 \text{ for simply supported} \\ 4 \text{ for clamped} \end{cases} \tag{3.9.1}$$

where $k_{b,eq}$ is the equivalent bending stiffness. Stating that $w_{c,u} = w_{c,eq}$ we can proceed to write :

$$\begin{aligned} w_{u,c} &= \frac{L^3}{48C\left(EI\right)^*} \\ \Rightarrow \left(EI\right)^* &= \frac{L^3}{48Cw_{u,c}} \end{aligned} \tag{3.9.2}$$

which means that our updated equivalent bending stiffness can be written as (insert equation (3.9.2) into (3.9.1)):

$$k_{b,eq} = \frac{1}{w_{u,c}} \tag{3.9.3}$$

where once more $w_{u,c}$ is obtained by simultaneously solving the system of equations due to the presence of one or multiple delaminations. In terms of a damage index affecting the entire bending compliance matrix d obtained from CLPT we rewrite the last row of equation (3.9.2) :

$$d_{11}^* = \frac{48Cw_{u,c}b}{L^3} \tag{3.9.4}$$

where we made use of the following relation:

$$\left(EI\right)^* = \frac{b}{d_{11}^*} \tag{3.9.5}$$

Once the equivalent bending compliance entry d_{11}^* is known we can define a damage index as:

$$d_i = 1 - \frac{d_{11}^*}{d_{11}} \tag{3.9.6}$$

Since for beam theory we only consider the (1,1) entry we can define the entire equivalent bending compliance matrix of the beam as:

$$\underline{\underline{d}}^* = \left(1 - d_i\right)\underline{\underline{d}} \tag{3.9.7}$$

If still needed, one can take the inverse of the left side of equation (3.9.7) to get the equivalent bending stiffness matrix D*. Finally, the force acting on the structure can be updated using the bending stiffness $k_{b,eq}$ from equation (3.9.3) according to:

$$P_{new} = k_{b,eq} w_{c,app} \tag{3.9.8}$$

where $w_{c,app}$ is the currently applied central beam deflection obtained from solving the impact problem (see section 3.3). It is important to mention that this update of the load acting on the beam always occurs just after the geometrical configuration of the beam changes due to newly created delamination sites or growing existing ones. The latter will be discussed in the following section.

3.9.2 Growth of delaminations

A delamination will start to grow if the computed value for the ERR is higher than the material's interlaminar fracture toughness. This value depends on the two ply orientations enclosing the interface [40] In this application we will assume that this value is constant. There are, however, differences concerning the mode of the delamination. Generally speaking mode II interlaminar fracture toughness has a higher value than mode I [46]. We discussed the differences in the contact formulation and what the consequences would be for the ERR in section 3.7. For the constrained contact formulation mode II occurs alone while there is the possibility for a mode I participation in the unconstrained contact formulation. It is theoretically also possible to obtain the individual ERR for the two different modes from the SIF computation [25]. However, due to the inherent oscillatory behaviour of the stresses close to the delamination tip, defining an accurate value for the individual components is not possible without extensive numerical resources. Finally, since any mode I contribution is likely to be minor we shall ignore it here and simply use the value for mode II interlaminar fracture toughness. Therefore we have delamination growth if $G > G_{cr,II}$.

Once a delamination is growing, one has to define by how much or how fast it is growing. In this work the following approach has been used. Once delamination growth has been determined, the length "a" of the delamination in question will be extended by a fix value "Δa". Here Δa has been set to one ply thickness. Since a change in length of a delamination constitutes a change in the geometrical configuration the load has to be updated as discussed above. Once the new load and delamination load are known the new ERR for this configuration can be determined. If the new ERR is still above the critical ERR it means that there is still sufficient energy for the delamination to grow further and the steps explained above will be repeated until the crack is arrested.

3.9.3 Updated stress field

Knowing the state of damage and the corresponding equivalent bending compliance in addition to the SIFs and the near crack-tip field emanating from the delamination tips enables us to compute an updated stress field to determine the next failure site based on the failure criterion proposed in section 3.6. The role of the contact stresses at this point is unclear. The approaches discussed in section 3.5 are only valid for undamaged sections since it is unclear how a delamination at an interface would influence the stress distribution. One could argue that if the two sublaminates do not separate the approaches might still be applicable. However, it is more probable though that the creation of a delamination site and the stress field emanating from it have a rather significant influence on the contact stresses, which is not likely to be captured by any analytical method and would have to be investigated with numerical tools. It was decided to use the contact stresses as determined in section 3.5 and superimpose those together with the bending stresses and the

singular stress field emanating from the delamination tip resulting in the final updated stress field. We will consider the example from Figure 3.13 with the values given in Table 3.2. The laminate layup is $[45/90/-45/0]_{3s}$ and the basic ply properties are given in Table 3.2 as well. From equation (3.4.5) we see that the transverse shear stresses only vary through the thickness and remain constant otherwise (see Figure 3.19). The other reason why the bending stresses are shown separately lies in the fact that contact and singular stresses are significantly larger and in the graphical superposition of the area of interest the bending stresses would not make an impact. This can be seen from Figure 3.21, where contour plots of the contact and singular stresses for σ_{xz} are shown individually and combined. Note that the contours are in Pascal and the same scale has been used for comparison reason. The x and y axes represent the height and length of the beam, whereby the midspan has been taken as the plot origin and only a quarter of the length has been plotted as beyond that point the influence of contact and singular stresses for this case vanishes. The impact site can clearly be identified from the first and third plot in Figure 3.21 as region of highly concentrated stresses.

Table 3.2 – values for example of Figure 3.13 and basic ply properties for IM7/8552 graphite epoxy composite

P	L	a_u, z_u	a_l, z_l
500 [N]	0.127 [m]	L/16 [m], $4t_{ply}$	L/8 [m], $16t_{ply}$

$E_{0,1}$	$E_{0,2}$	$G_{0,12}$	$v_{0,12}$	t_{ply}
144.8 [GPa]	8.3 [GPa]	5.8 [GPa]	0.32 [-]	0.19e-3 [m]

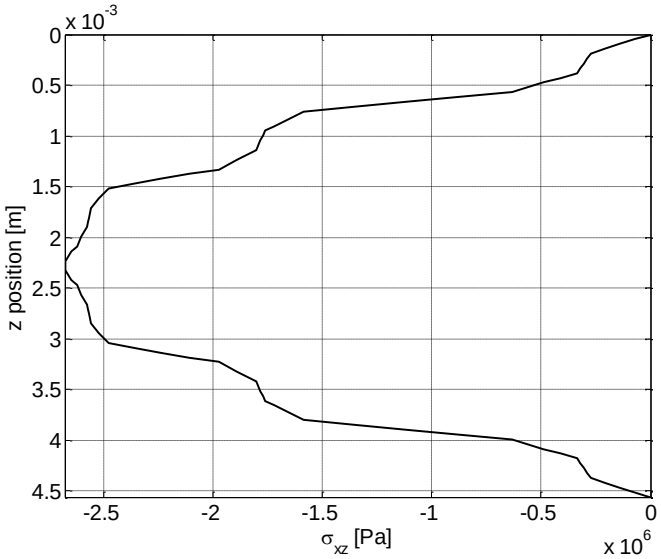


Figure 3.19 - Transverse shear stress distribution due to bending (for example in Figure 3.13)

We know from the derivation of the contact and bending stresses that the boundary conditions are met. By inspection one can see, however, that this is not the case for the singular stress fields. Considering the order of magnitude of the stress levels the locations immediately above and below the delamination tips seem to show residual stresses at the boundary which are not negligible. To obtain a more detailed view the stresses along the boundaries have been plotted in Figure 3.20. The peaks in the stress values corresponds well with the location of the delamination tips. The

reason for these high residuals can be explained as follows. During the derivation of the SIF the boundary conditions on top and bottom of the beam were set to be traction free (see Suo & Hutchinson). The formulation of the singular stress field does not allow for the boundary conditions to be accounted for nor does it contain corrections for finite size. This lead the research to expressions for the stress field around a crack containing more than just the singular term with the aim to find a stress field such that the boundary conditions will still be met. There are two alternative approaches which allow for the inclusion of higher-order terms, namely the Williams stress function and the complex variable formulation due to Muskhelishvili (see [22]) Both approaches have been attempted, however, the results were not the expected ones and the implementation was certainly less flexible with solutions for cracks lying at interfaces not yet considered. For the sake of good documentation these attempts have been briefly summarized in Appendix (not yet written). Taking all of this into account it has been decided at this point that the current solution regarding the singular stress field will be used. The overall approach that has been followed allows for future researchers to take this framework and substitute a more suitable module for the computation of the SIF.

Once the updated stress field is known the failure criterion can be applied again to check whether the next delamination has been initiated. With this the loop shown in Figure 3.1 is closed. At this point the impact simulation can be carried out until the maximum deflection is reached and the internal damage state can be extracted. In the following section this model will be put to use and evaluated based on a comparison with published results.

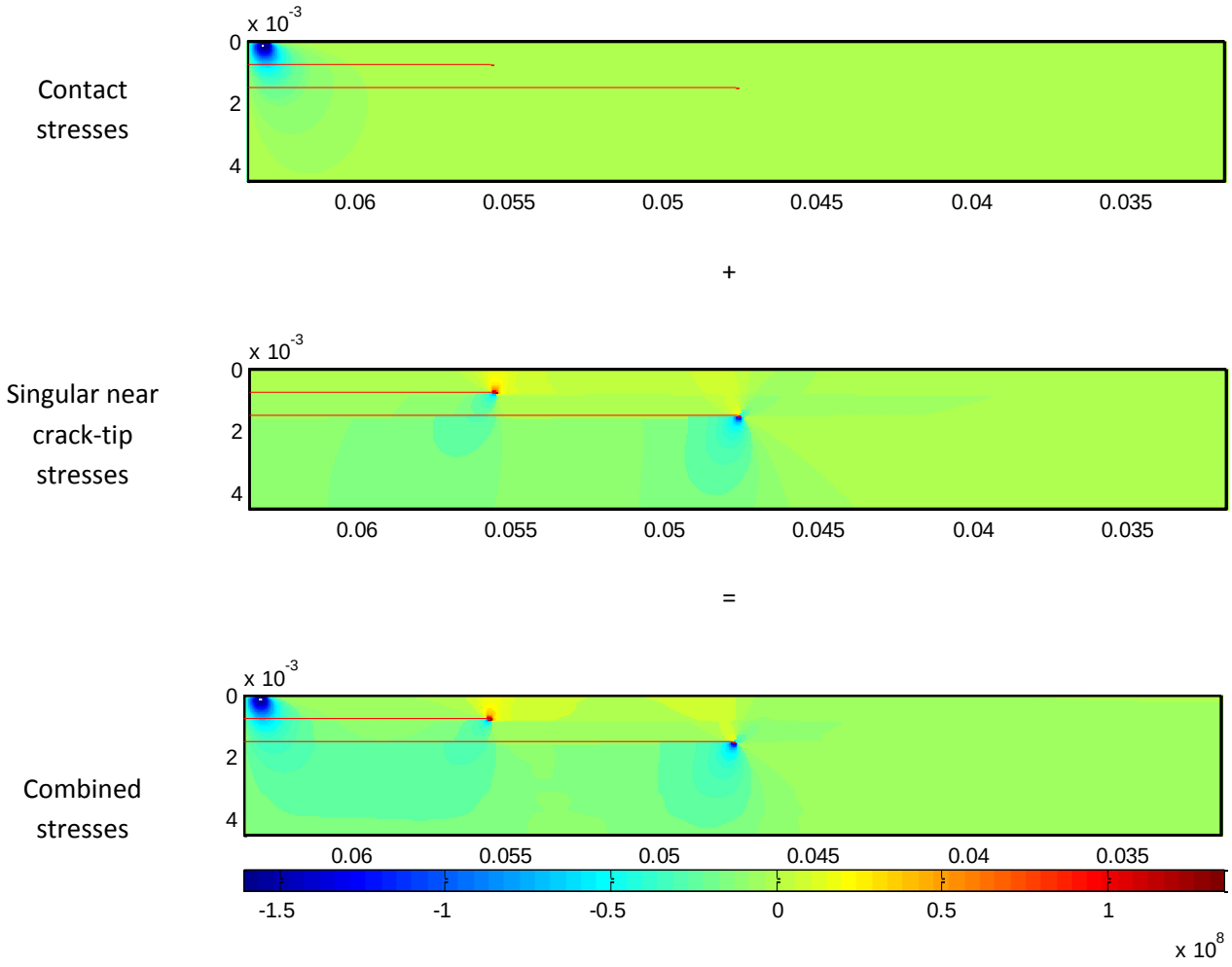


Figure 3.21 - contact and singular stress superposition for the example from Figure 3.13

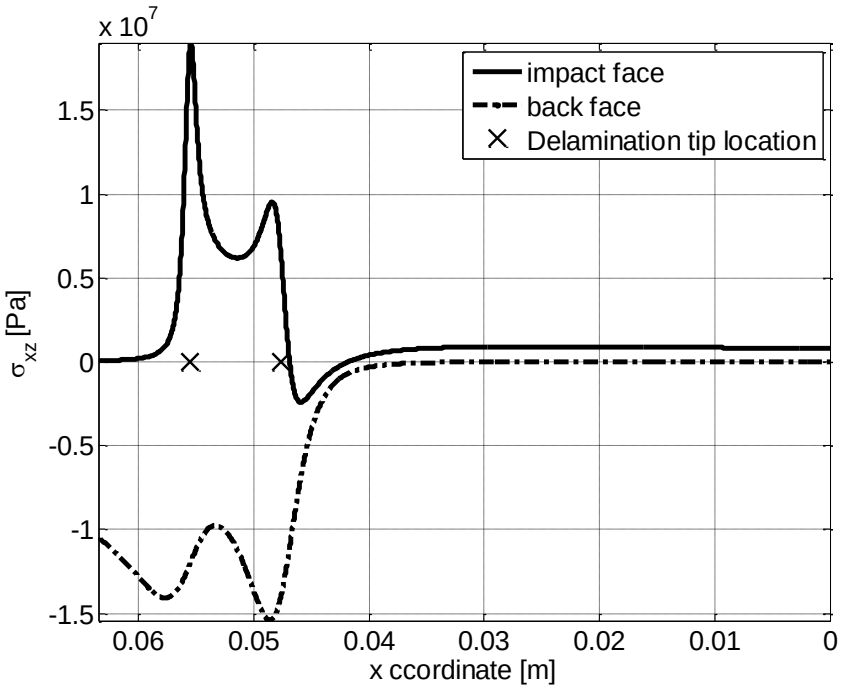


Figure 3.20 - Boundary stresses due to singular stress field

4 Model application

The model that has been build and discussed in the previous section will now be used to simulate several impact events, the results of which will be compared in terms of the damage created and the structural capabilities of the laminate after the impact.

4.1 Simulation set up

For the comparison of the model the paper by [47] will be used. They have investigated the effect of laminate stacking sequence on damage tolerance for centrally impacted composite plates. The material that they used was graphite epoxy IM7/8551-7 [48] has already been introduced and is summarized in Table 4.1. Certain properties could not be found in the data sheet and had to be assumed. The critical ERR has been reported in [46] for IM7/8552, which is assumed to be the same for the IM7/8551-7 system.

Table 4.1 - IM7/8551-7 basic ply and strength properties

Variable	$E_{0,1}$	$E_{0,.2}$	$G_{0,12}$	$v_{0,12}$	t_{ply}	X_t	X_c	Z_c	$S_4 = S_5$	G_{cr}
Value	144.8	8.3	5.8	0.32	0.19e-03	2757	1620	200[4]	100	1138[5]
Dimension	GPa	GPa	GPa	-	m	MPa	MPa	MPa	MPa	N/m

The following laminate layups will be used for the simulation:

Table 4.2 - test layups

LAM1	$[45/90/-45/0]_{3s}$
LAM2	$[45_3/90_3/-45_3/0_3]_s$
LAM3	$[30/60/90/-60/-30/0]_{2s}$
LAM4	$[30/60/90/-30/-60/0]_{2s}$
LAM5	$[45/(90/-45)_3/(0/45)_2/0]_s$
LAM6	$[45/(0/-45)_3/(90/45)_2/90]_s$
LAM7	$[45_2/90_2/-45_2/0_2]_{2s}$

All layups have 24 plies, except of LAM7 which has an extra 8 plies. The reason for choosing this layup has to do with the fact that the effect of ply group thickness can be demonstrated very well using LAM1, LAM2 and LAM7. The length and width of the plate simulated were 12.7cm and 7.62cm and correspond with the impact specimen size reported in Dost et al. The impact test they conducted featured a 5.44kg steel sphere with a radius of 0.8cm. Every layup has been simulated at three energy levels: 15, 20 and 25 J. Although it has been previously stated that 20 J would be the upper limit for the model considered it would imply that for certain laminates only a single comparison point would have been possible. The energy levels chosen here do not correspond

[4] Taken from [40] (table 6.1)
[5] Taken from [46] for IM7/8552

exactly with those reported in [47], however, as these were not entirely fixed across the range of laminates it was chosen to select energy levels which would cover a big range of the tests conducted by . Knowing the impact energy and mass the impact velocity needed for the impact simulation was easily obtained. In order to see what influence the choice of contact formulation has all simulations were run with either constrained or unconstrained contact.

The model from the previous section has been programmed with MATLAB®. The entire source code including a flow chart with brief descriptions explaining the whole programme has been provided in Appendix (*only raw at the moment*). Different possibilities with respect to how the programme behaves when one or multiple delaminations occurred were considered. This discussion can be found in Appendix XX. At this point it is important to know the settings of the simulations and any possible implications thereof. With respect to the SIF it has been decided that the singular near crack-tip stress field at every delamination tip will be used in the simulation to update the stress field in the damaged beam. The other major consideration concerned the continued application of the failure criterion at an interface which has been found to fail at a previous load. For the simulation shown here it was decided that even though a particular interface was "acting" as a delamination already, the failure criterion proposed would still be used on that interface to check whether failure would be detected at another distance away from the impact site. This implies that even though the delamination does not grow from a fracture mechanics point of view (G > Gcr) it might still "grow" as the stresses ahead of the tip are large enough to render the failure criterion larger than unity. Only if the delamination is actually growing as discussed in chapter 3 will this check stop. While this way of looking at delamination growth is different than what has been discussed in chapter 3 it is still based on the concepts of LEFM via the SIF and the corresponding singular stress field emanating from the delamination tip. Further the method applied by Choi and Chang [36] is very similar in that they also use stress-based criterion to determine the extent of a delamination.

Finally, it has to be mentioned that the comparison with the results published by Dost, Ilcewicz [47] can only be in a qualitative manner, since in they conducted experiments on plate-like structures with delaminations that were fully enclosed by surrounding material. The current model uses beam-like structural analysis with delaminations that span the entire width of the beam, which in this comparison equals the width of the plate.

4.2 Impact damage

After impact testing one would like to inspect the internal damage state created. Normally, this is done with some form of non-destructive testing, such as pulse-echo scans. These scans then reveal the state of damage when viewed from above. Alternatively, one can cut the laminate at the impact site to expose the internal structure and inspect it under a microscope. Interesting imagery of both techniques can be found in [40]. Delaminations due to impact are found at nearly every interface and most of the time in an unsymmetrical arrangement, i.e. that smaller delaminations are located closer to the impact site whereas the larger delaminations are placed near the back face. In Figure 4.1 the internal damage state of $[45/90/-45/0]_{3s}$ is shown for two impact energies, following the unconstrained contact formulation. Due to symmetry only half of the structure is shown, whereby the line of symmetry coincides with the line of action of the load P indicated. In Figure 4.1 a) it can be observed that the damage has not yet spread beyond the mid-line of the layup. This is in stark contrast to the observations made in [47]. The damage state in Figure 4.1 b) is more extensive, however, the expected cone pattern can still not be observed. In Figure 4.1 c) the corresponding load-displacement graphs are shown. The drops in the impact load result from either a growing

delamination or the creation of a new delamination. Towards the end of the 20J impact there is a huge drop in load which corresponds to the creation of the two delaminations right underneath the midline in Figure 4.1 b). This observation has been made with other simulations as well. There are two possible related reasons for this behaviour. Every time a new delamination is created at an interface there is a jump increase in displacement due to a unit load and accordingly a drop in load due to assumption of applied displacement. When a new delamination is defined at an interface below the mid-line this jump is more intense in magnitude resulting in rather large drop in force. While the method of solving for the central deflection of a multiply delaminated beam is correctly applied this large drop seems to be a direct consequence of the applied beam theory. The other reason connected to this concerns the assumption of applied displacement. While this assumption serves to simulate test set ups it will not be encountered in real life, where more appropriately a mix of applied displacement and force would occur.

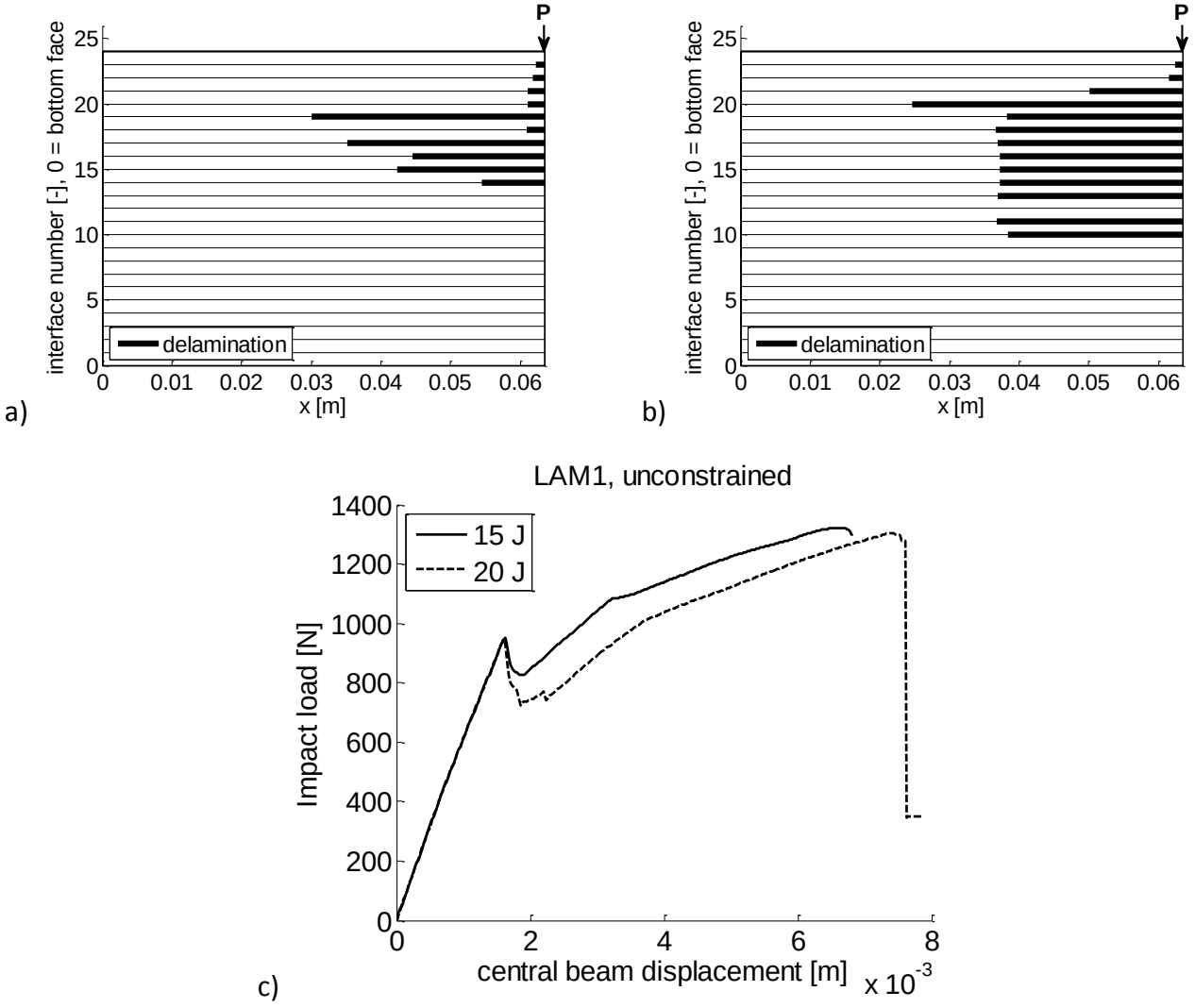


Figure 4.1 - Internal damage state of $[45/90/-45/0]_{3s}$ after an 15J (a) and after 20J (b) impact. Load vs displacement graphs (c)

In their paper Dost, Ilcewicz [47] only use the non-destructive technique to inspect the internal damage state, since they intended to test the impacted specimen afterwards. This means that the first comparison will be based on the damage diameter due to impact. Delaminations due to impact are not penny-shaped as often assumed but obtain a shape that resembles a peanut (see [49]) whereby the longitudinal axis is aligned along an angle defined by the ply orientation encompassing

the delamination. When several of these peanut-shaped delaminations are viewed from top a circle can be described which contains all of the damage. The diameter of that circle is called the damage diameter and used as measure for the impact resistance of a laminate. Since the current model is based on a one dimensional beam model, the longest delamination in the stack will be used for the comparison. For the two cases of unconstrained and constrained contact these values have been summarized in xx a) and b) and compared with the values reported for the damage diameter in [47] in c). The overall trend shown in Figure 4.3 c) and reported in [47] describes a larger damage diameter for increasing impact energies. The same observation can be made for LAM2 irrespective of the contact formulation used. The relative position of LAM2 with respect to the other laminates also corresponds well with [47]. It is mentioned that for this laminate large planar delaminations occur due to the increased ply group thickness compared to LAM1 and LAM7. Figure 4.3 shows the internal damage state for LAM2 and LAM7 for a 15J impact event. Together with Figure 4.1 a) this observation can be confirmed. The predicted delamination diameters correspond in terms of order of magnitude, but almost without exception are several times larger than the test results. The reason for this and the lack of trend in the predicted results could lie in the fact that the current model is not able of modelling two-dimensional delaminations and assumes delaminations to be through the width of the beam. Further, the fact that other damage modes are more dominant after impact levels of 20J ([40]) and not considered might provide another explanation as to why the longest delamination only might not work as damage indicator.

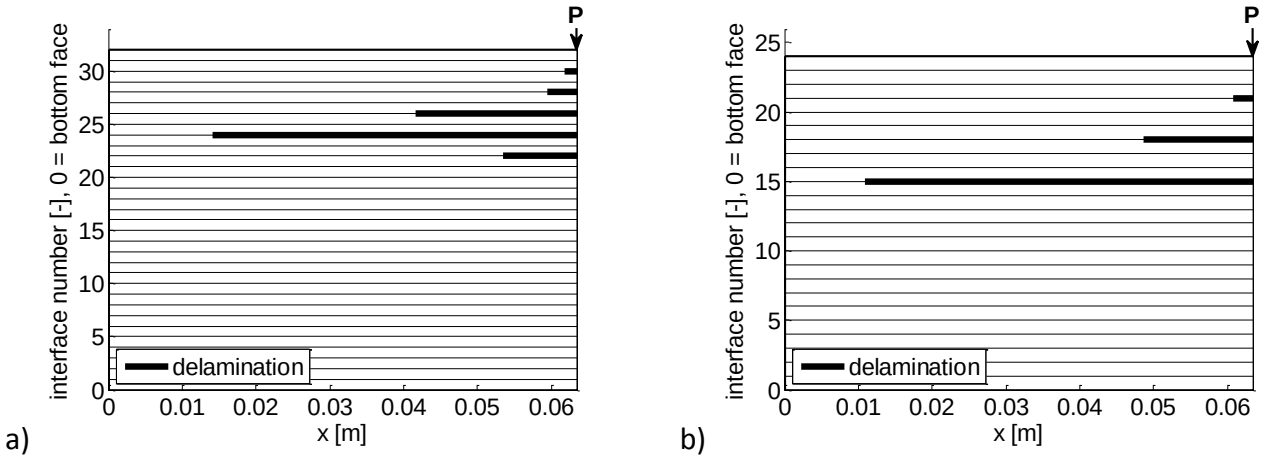
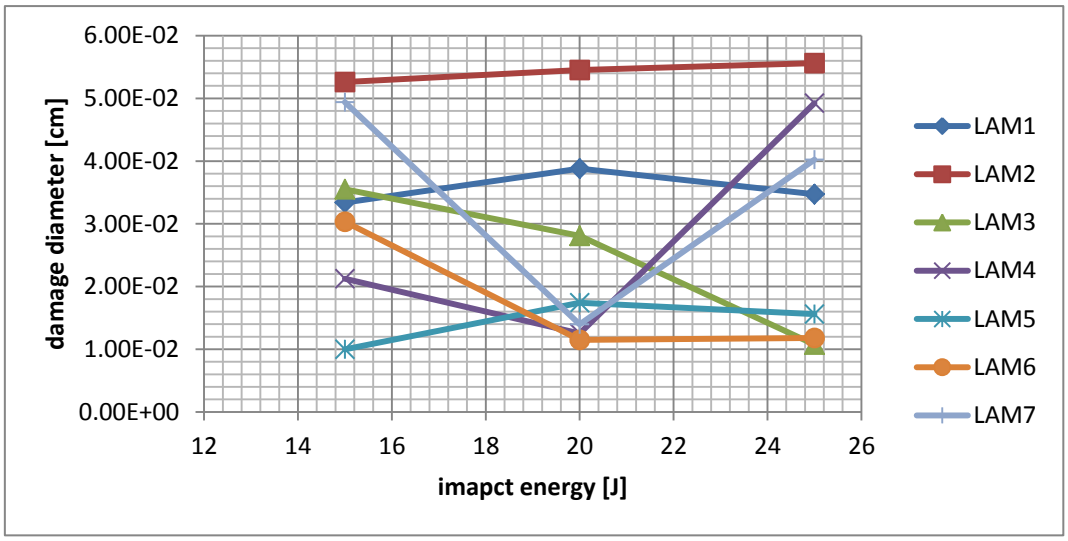


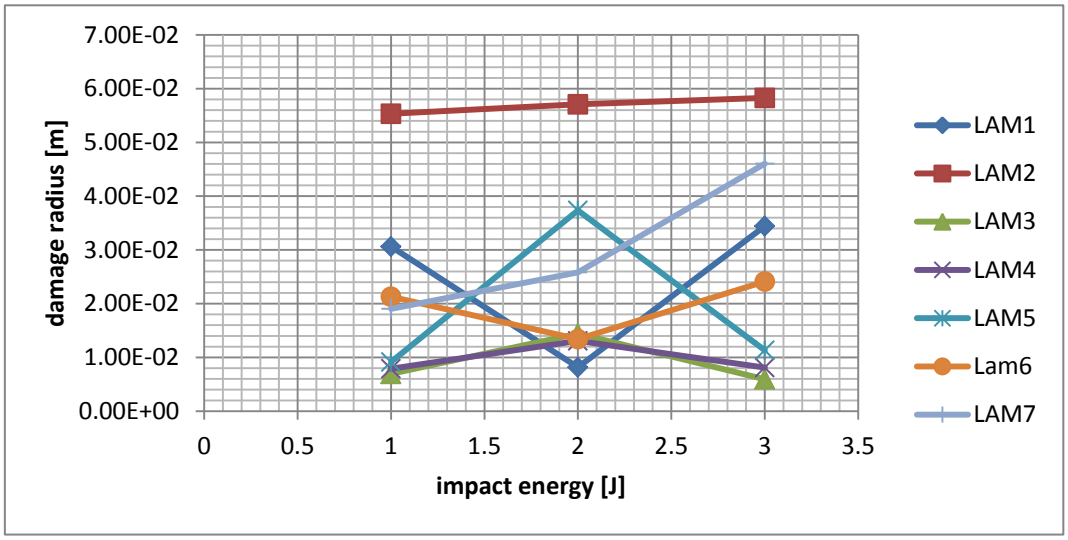
Figure 4.2 - internal damage state of a) $[45_2/90_2/-45_2/0_2]_{2s}$ and b) $[45_3/90_3/-45_3/0_3]_s$ after 15J impact

LAM7 and LAM5 in Figure 4.3 a) and b) stand out from the rest by the seemingly random behaviour. This behaviour, however, can be explained when looking at the internal damage state of these lamaintes for the corresponding contact formulation. Those are shown in Figure 4.4. The top row, which represents LAM5 is difficult to interpret, as it is not clear which mechanism in the model would keep the delaminations in the upper half for a higher impact energy, when before the delaminations were spread through the entire stack. The situation is reversed for LAM7 (Figure 4.4 bottom row). Again though it is unclear why in the bottom left graph LAM7 features a relatively long delamination rather than several through the thickness delamiantions as shown in the bottom middle graph. It is believed that this has to do with the fact that for different impact energies the force that would lead to a new delamination might occur just before further growth of an existing delamination. With the occurrence of a newly delaminated interface this growth might just be prevented from happening leading to the state that can be seen in Figure 4.4.

a)



b)



c)

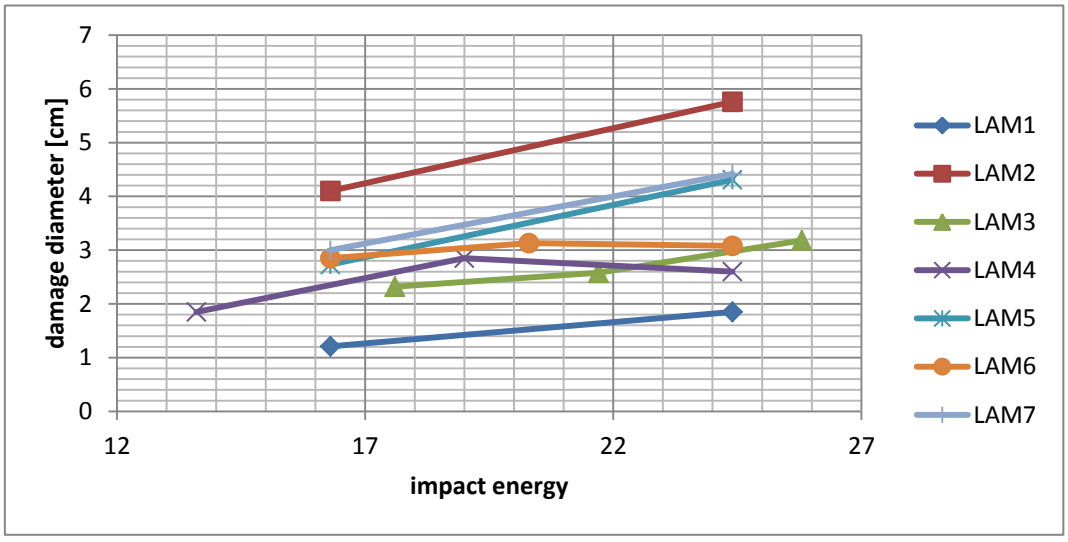


Figure 4.3 - comparison of impact damage: unconstrained (a), constrained (b) and data taken from [47]

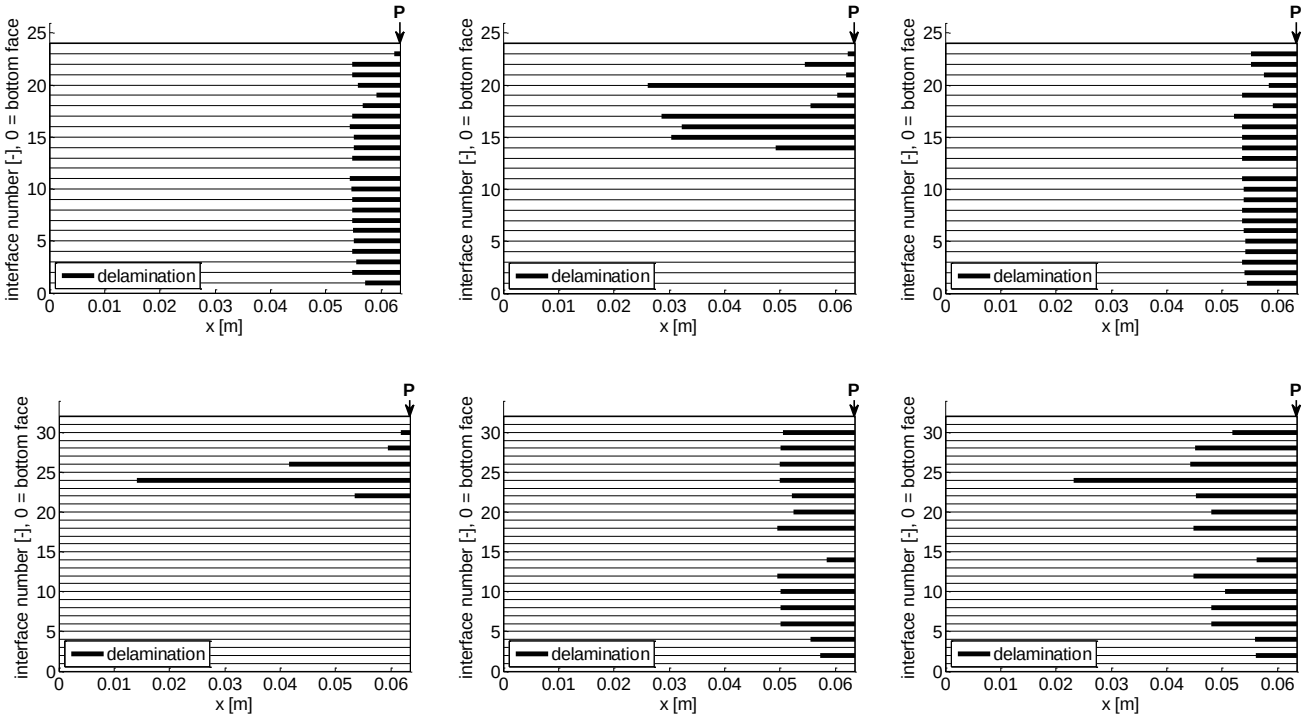


Figure 4.4 - internal damage state for $[45/(90/-45)_3/(0/45)_2/0]_s$ (top row) and $[45_2/90_2/-45_2/0_2]_{2s}$ (bottom) for various impact levels (left to right: 15J, ,20J and 25J)

Although the current model in itself will not be able to predict a two-dimensional damage by a single run, a compound solution can be obtained. This involves running the same simulation twice: once with the original layup and plate orientation and a second time with the plate dimensions and layup rotated by 90 degrees (see Figure 4.5). This will allow for the definition of a damage footprint in the form of an elipse (see Figure 4.6), by defining the largest delamination from the original analysis as one axis of the elispe and the largest delamination of the rotated analysis as the other exis of the elipse. The analysis stays the same only now the original plate's width has to be used as

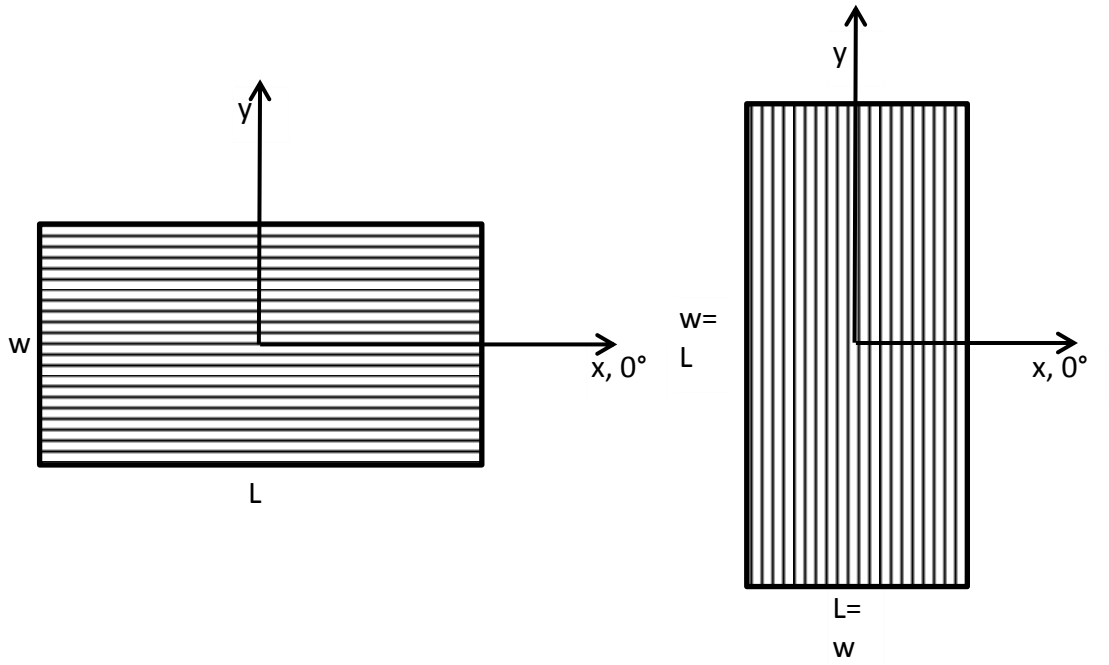


Figure 4.5 - rotation of plate and layup

length and vice versa. Also the layups change by 90° as observed in Figure 4.5. Originally the 0°-fibres are aligned with the x axis, however, when the plate is rotated the axis system stays and the 0°-fibres now become the 90° fibres.

This approach has been applied to LAM1 and LAM7 for a 15J and 20J impact. Once the damage envelope has been defined the area of the ellipse can be computed and from that value the diameter of a circle with equivalent area can be calculated. The results are summarized in Table 4.3 and a visualization for the 15J event is provided in Figure 4.6 and Figure 4.7.

Note that this analysis has been carried out using the unconstrained contact formulation. The prediction for LAM1 is still a factor of 3-4 higher than the results published in [47] and for LAM7 there is still a decrease in the damage diameter for an increase in impact energy. At this point more analyses are necessary to provide a thorough explanation for this behaviour. When using this approach one has to be aware of the fact that a beam analysis is conducted on a specimen where the length is shorter than the width.

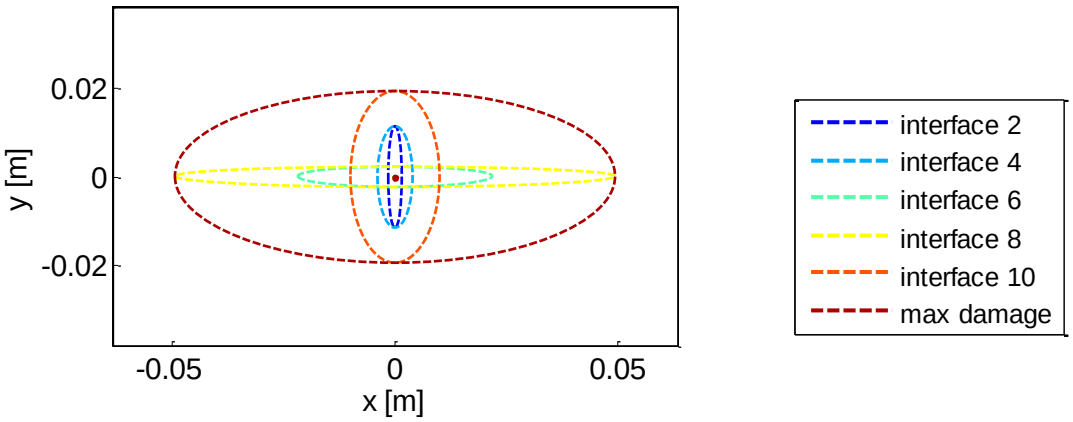


Figure 4.6 - planar view of elliptical delaminations for $[45_2/90_2/-45_2/0_2]_{2s}$ obtained by rotating the analysis frame

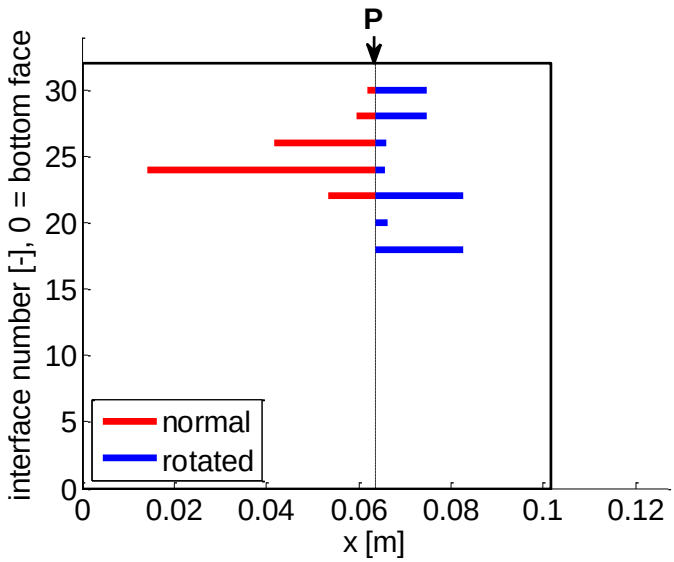


Table 4.3 - equivalent diameter [cm] from two-dimensional damage analysis

	15J	20J
LAM1	5.15	6.35
LAM7	6.16	3.54

Figure 4.7 - internal damage state for $[45_2/90_2/-45_2/0_2]_{2s}$ in normal

4.3 Damage tolerance and optimization criterion

Next to damage resistance, which indicates how well a laminate can contain damage, the concept of damage tolerance of a laminate is as important. It describes the ability of a laminate to cope with loading after impact has taken place. Normally, CAI tests are performed to assess this property since delaminations are mostly effective during compressive loading. Delaminations have a decreasing effect on compressive load carrying capabilities. Since carrying out CAI analyses was not part of the current work another way had to be found to compare the results of the simulations with the results published in [47]. A criterion was needed that would be able to rank the laminates in a similar fashion to CAI results, i.e. it should combine the certain aspects of the internal damage state and structural performance of the delaminated beam such that it would reflect a CAI analysis and can be used as criterion in an optimization process. To this end the following three items will be combined in a criterion: the equivalent bending stiffness, the longest delamination and the sum of delaminations. The latter two will be weighed with respect to their position within the stack. The function is then formulated as follows:

$$f = \frac{K_{eq}}{K_{0,b}} + \frac{L}{a_{\max}} \frac{\left| h_{\max} - h_0 \diagup 2 \right|}{h_0} + \sum_i \frac{L}{a_i} \frac{\left| h_i - h_0 \diagup 2 \right|}{h_0} \tag{1.1.1}$$

A higher function value means a better laminate after damage, i.e. shorter delaminations placed towards impact and back face of the laminate will result in a better equivalent bending stiffness and consequently in better CAI result. Since in bending or compression analysis the thickness of the structure almost always appears in a cubed fashion, a delamination further away from the midplane will be less detrimental to compressive loading than a delamination in the middle.

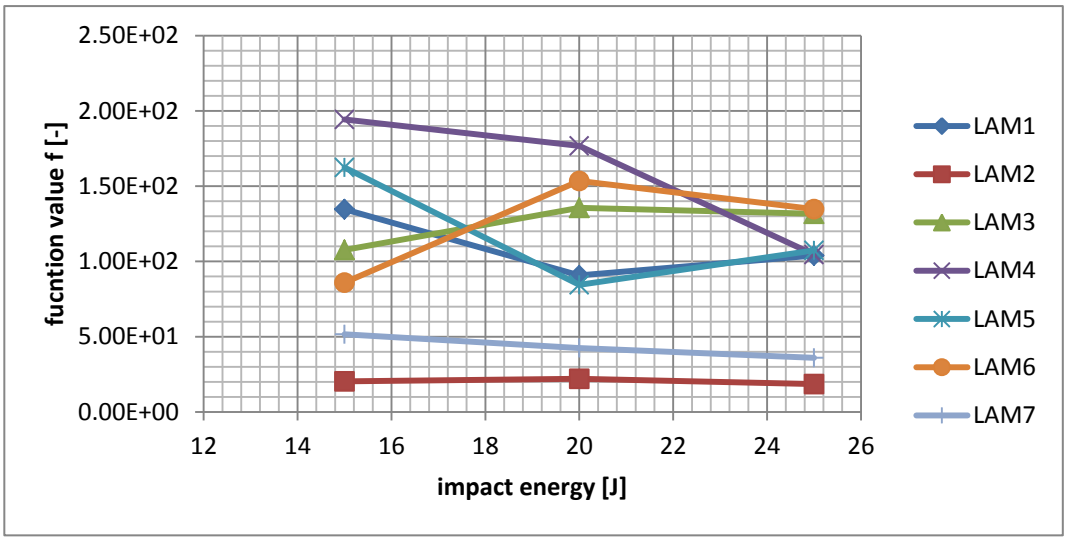
The results of the simulations in terms of this function value are summarized in Figure 4.8 a) and b) for the unconstrained contact formulation. In the same figure the results for the CAI experiments from [47] are presented as well (Figure 4.8 c)). To start with we focus on LAM1, LAM2 and LAM7 where the same trend for these laminates can be observed as reported in [47]. As the ply group thickness is increased from one to three plies the function value decreases, similar to the reported CAI values for these laminates. Also the fact that LAM2 obtains the lowest score agrees well with the tests conducted. The reason for this was based on the fact that when plies are grouped together larger planar delaminations would occur, which was already mentioned before and can be observed by looking at Figure 4.1 and Figure 4.2.

From Figure 4.8 c) it can be seen that LAM3 performs best in the CAI despite the overall larger laminate thickness of LAM7. However, the predictions for LAM3 and LAM6 both show very irregular behaviour for both contact formulations. This is why these two laminates will be looked at in more detail trying to explain what causes this seemingly random behaviour. This will be done for the constrained contact formulation (Figure 4.8 b)).

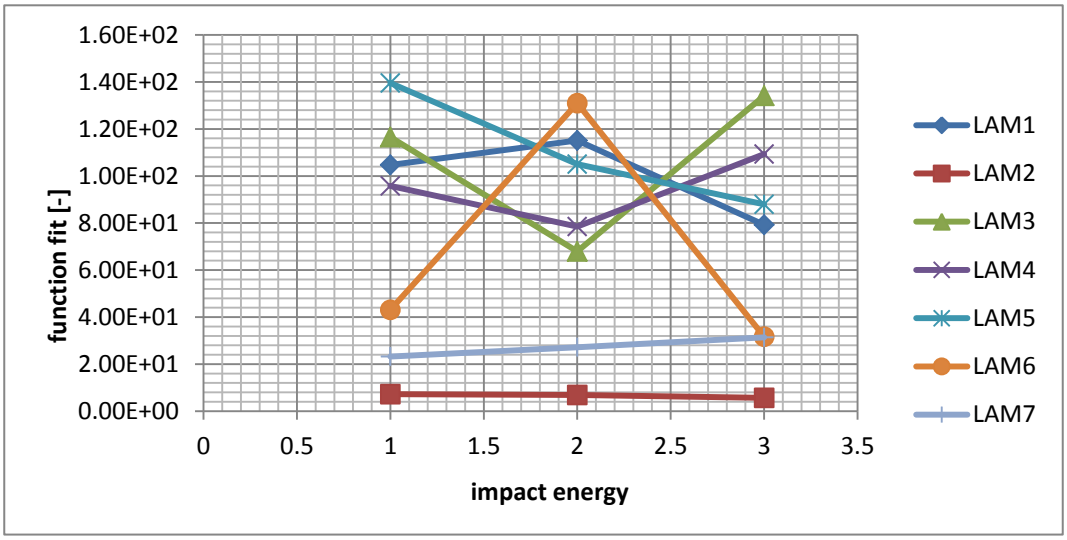
4.4 Discussion of the results

In this section the results of the simulations will be discussed. A more detailed dissemination of LAM3 and LAM6 for the constrained contact formulation precedes a more general discussion of the entire model.

a)



b)



c)

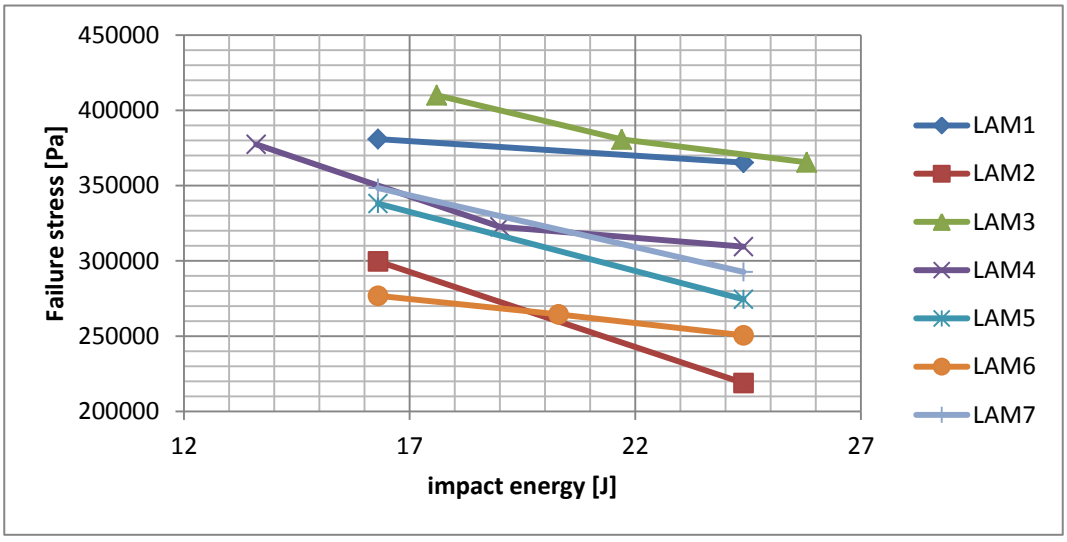


Figure 4.8 - Damage tolerance comparison using (1.1.1) and CAI results from [47]. Unconstrained (a) and constrained (b) contact formulation

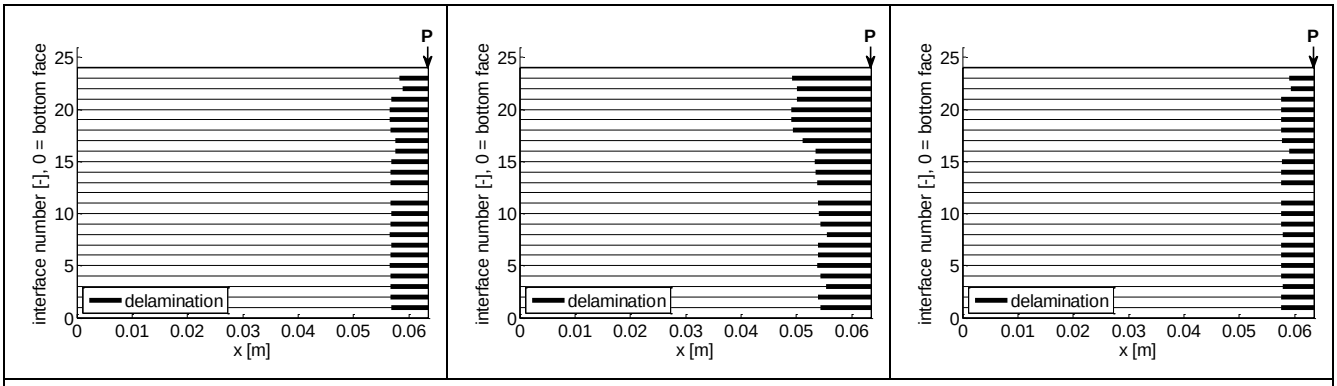


Figure 4.9 - internal damage state for $[30/60/90/-60/-30/0]_{2s}$ for various impact levels (left to right: 15J, ,20J and 25J)

The three internal damage configurations after impact for LAM3 are shown in Figure 4.9.The change in function value for LAM3 from 15J to 20J makes sense when looking at the internal damage configuration. Overall larger delamination sites lead to a decrease in equivalent bending stiffness as well as the second and third term in equation (1.1.1). While this is as expected and does not pose any questions the increase in the function value for the 25J impact to a value above that for 15J is certainly explainable by looking at the internal damage state, however, does not seem to follow any particular logic. The delamination interfaces are the same as for the lower two energy levels, however, contrary to what one would expect the lengths of the delaminations stayed below those of the 15J impact. It is noticed that the creation of each new delamination in the case of an 25J impact occurs earlier than in the case of a 15J impact. This has to with the fact that the impact load necessary to produce high enough failure stresses is reached slightly earlier for the 25J impact than for the 15J impact. It is possible then that due to the chosen time increment the simulation carried out in this unexpected manner.

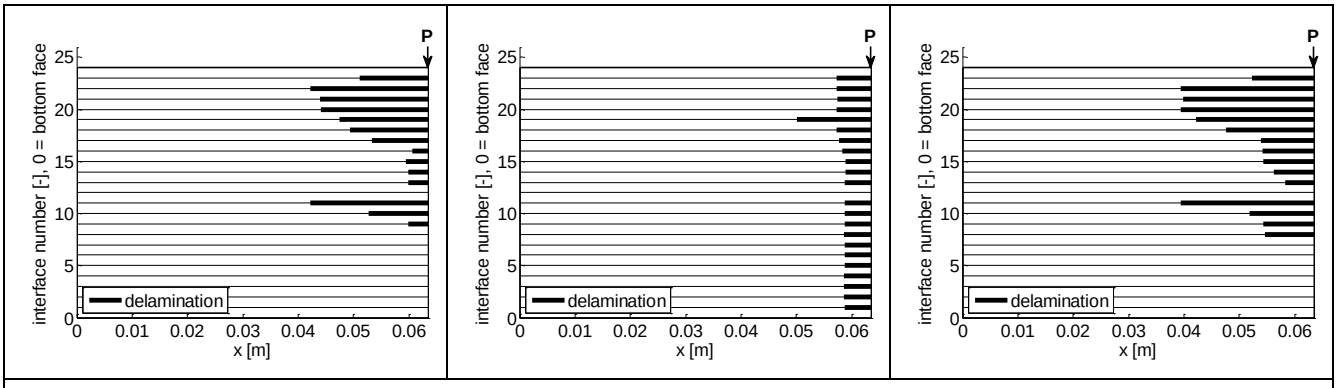


Figure 4.10 - internal damage state for $[45/(0/-45)_3/(90/45)_2/90]_{2s}$ for various impact levels (left to right: 15J, ,20J and 25J)

The internal damage states for LAM6 for the three impact levels are shown in Figure 4.10. Again the 20J impact appears to fall out of the sequence, because it presents a rather different damage state than the 15J or 25J impact. The parameters for the function evaluation for all three cases are given in Table 4.4.

Table 4.4 - terms of the function to evaluate damage tolerance for $[45/(0/-45)_3/(90/45)_2/90]_{2s}$

E_{imp}	$\frac{K_{eq}}{K_{0,b}}$	$\frac{L}{a_{max}}\frac{\left\lvert h_{max}-h_0/2\right\rvert}{h_0}$	$\sum_i \frac{L}{a_i}\frac{\left\lvert h_i-h_0/2\right\rvert}{h_0}$	f
15	5.8705e-002	2.4836e+000	4.0584e+001	4.3126e+001
20	5.4437e-003	2.7644e+000	1.2821e+002	1.3098e+002
25	4.1615e-002	2.1940e+000	2.9403e+001	3.1638e+001

The main differences are observed in the equivalent stiffness term and the summation of the delaminations. According to the predictions, having delaminations across the entire stack results in a lower equivalent bending stiffness. The difference with respect to the other two impact events is one order of magnitude. The summation of the delaminations obtains a value which is one order of magnitude higher mainly due to the increased number of delaminations. However, when comparing the orders of magnitude of these two terms it becomes apparent that the summation term overshadows the stiffness term by several orders of magnitude, which is why the lower performance in terms of stiffness for the 20J impact has hardly any effect on the function value. The reason for the stiffness term to be so low can again be explained by the way the central displacement of the beam is solved for. By including judiciously chosen factors one can aim to bring all terms to the same order of magnitude such that a more reliable comparison can be carried out in case of different numbers of delaminations.

The application of the model to published results is inconclusive at this point. Some aspects, such as the relation between LAM1, LAM2 and LAM7, as mentioned in [47] have been captured well by the model while for example the damage created did not correspond well with experiments [47]. The mayor difference between the model and the results from the observations is the missing delaminations in the lower part of the laminates for a variety of the simulations carried out in connection with the cone-shaped damage state. This might be due to the fact that preceding failure modes such as shear and matrix cracks and fibre breakage were not considered as delamination initiators and other failure modes in the model. It is plausible that under loads that occurred during the simulation such cracks might have formed in the lower part of the laminate. The other factor that most certainly has an influence is the fact that the structure considered in this model might be ill-suited for comparison. Having delaminations spanning the entire width of the beam, which takes the value of the impact specimen used in [47], will have a profound impact on both the deflection and thus stiffness of the delaminated beam and the ERR and SIF values.

Assuming applied displacements has led to the continuous update of the force acting on the structure. While drops in the force displacement curve indicate a change in the structural behaviour due to internal damage, large drops such as shown in Figure 4.11 for the graph belonging to the 20J impact are unlikely to occur in real life. A real impact event is neither purely governed by applied displacement nor by applied force but rather by a mix between the two. It can be observed that the sudden drop in load corresponds with the creation of delaminations in the lower half of the laminate. The same observation can be made for other combinations of laminate layup and impact energy. Again, a possible explanation of this phenomenon lies in the fact how the central displacement of a multiply delaminated beam has been obtained, which serves as input for the force update. It is highly probable that the exclusion of shear deformations leads to unrealistic

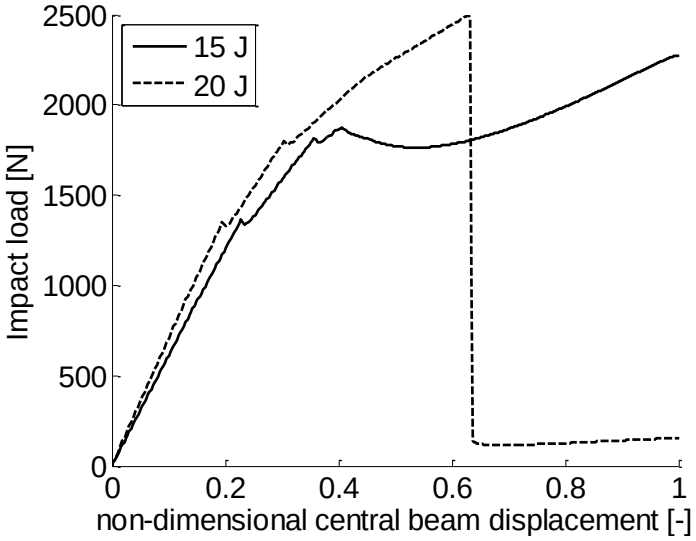


Figure 4.11 - Force vs. displacement graphs for $[45_2/90_2/-45_2/0_2]_{2s}$ layup

displacement values which cause the load to drop so significantly. No clear observation can be made about the differences in constrained vs. unconstrained contact formulation, which is to be expected as both are neither wrong nor correct but simply attempt to represent reality from two extreme points.

Finally, it is impossible at this point to judge how influential the adoption of the LEFM concept was. While the computation of the ERR is straightforward for a given displacement field, the SIFs, and the way they have been computed for each delamination tip, still left some questions. Determining the extent of delamination damage based on the interlaminar stress values has been done before, however, the correctness of the stress field surrounding the tips is crucial for a successful application of such a criterion.

5 Conclusions and recommendations

In this chapter the conclusions with respect to the work performed will be discussed. Finally, recommendations for future work will be included as well.

5.1 Conclusions

The first thing to ask when trying to conclude on something is whether the goal that has been set in the beginning has been achieved, partly or wholly. To this end the research objective for this project will be given again:

To develop a (semi-)analytic model for multiple delaminations in an impacted composite plate by obtaining an accurate representation of the stress field near the impact zone, using failure criteria in combination with fracture mechanics concepts to determine conditions for delamination initiation and growth, and to investigate the interactions between delamination zones.

Overall it can be concluded that this objective has been accomplished. A model has been created, which:

- based on a quasi-static approach computes the central displacement of a beam,
- predicts the stress state accurately prior to first damage initiation, using either semi-infinite plates or plates with finite thickness
- predicts the first and all subsequent delamination using a modified stress-based delamination initiation criterion
- compute the deflection of beam segments or sections depending on the contact formulation chosen
- computes the ERR for every delamination tip individually based on the displacement field
- uses a published approach for the determination of the SIF at the delamination tip for the computation of the near crack-tip singular stress field
- updates the impact force when delaminations are growing or new ones created

When developing the model care has been taken to keep it as modular as possible so as to make it adaptable for future use. By simply replacing certain modules, performance in terms of time and accuracy should be achievable.

This model has been used to simulate impact events of increasing impact energy on quasi-isotropic laminates using either the unconstrained or constrained contact formulation. The obtained results have been compared with published ones in terms of damage resistance and damage tolerance. In the case of damage resistance, which is measured in terms of the damage diameter, the predicted results have been found to be in reasonable to poor agreement, based on the considered laminate. An attempt has been made to use a rotated analysis to determine a two-dimensional state of damage. Using this data the predicted damage diameter was still 3 to 4 times higher than experiments have shown. This result is encouraging considering the fact that a one dimensional structural element is used with through the width delaminations. The absence of delaminations in the lower half of the laminate for certain configurations has been noted. In terms of damage tolerance no quantitative comparison was made but rather a qualitative comparison based on a criterion which would be fit for optimization purposes. Again encouraging results for a series of

laminates have been obtained, while the proposed criterion seemed to be ill-adapted for other laminates.

During the building of the model and the analysis of the results three main issues have been encountered. The first one concerns the computation of the stress intensity factor at the delamination tips. The published approach [1] has been used by a variety of researchers, however, when implemented in connection with the singular near crack-tip field excessive residual transverse stresses were found at the top and bottom boundaries of the beam. Secondly, the solution invoked for solving the problem of a multiply delaminated beam is based on LBT, neglecting shear deformations. There are situations throughout the simulation process where not all of the conditions necessary for the application of this theory are met, This might yield unrealistic displacement values, which subsequently lead to large drops in the impact load. This last remark leads to the third and last main issue. During the simulation applied displacements have been assumed, which is often used in laboratory environments. The actual impact event though cannot be captured by this simplifying assumption and rather features a combination of applied displacement and force.

This implies that next to the encouraging results seen already there is still plenty of work to be done solving this particular problem. Some useful recommendations that might help future researchers with this task are given below.

5.2 Recommendations

- Throughout the project interesting ideas with respect to improving the current model occurred which will be presented here as recommendations for future work:
- Further investigation into the Williams stress field or the complex variable formulation for cracked configuration, with emphasis on multiple interface cracks, is desired, as it presents a method for computing the entire stress state, including SIF, in a cracked body, whilst considering boundary conditions.
- The ERR definition by Hwu & Hu and Banks-Sills using the Stroh Formalism are potentially the way forward for composite LEFM. However, they rely on FE input for the computation of the SIF (Banks-Sills uses M-integral in combination with FEM). This probably is the most accurate when it comes to defining SIF at an interface between two composite plies, however, it lacks simplicity when trying to implement it in the current framework.
- The exclusion of matrix and shear cracking as preceding and separate failure modes has to be re-evaluated. They form an integral part of the damage mechanism and it might therefore be crucial to incorporate them, if not necessarily from a structural point of view but from a phenomenological point of view.
- Proper in-situ strength parameters as well as fracture toughnesses should be used. Both are dependent on the angle difference of the interface. Also a mix-mode delamination growth criterion in cases where both modes occur is advisable.
- The current LBT needs to be updated to incorporate shear deformations and possibly even root rotations. Further, sublaminates possess coupling effects which also influence the overall deflection of a multiply delaminated beam.
- Use the spring-contact model to accurately define the interaction between beam segments for improved deflection prediction.

- Reconsider the assumption of applied displacement. A combination between applied force and applied displacement will be able to represent the actual behaviour more closely. In this context it might be even advisable, to consider using a dynamic approach wherein the stiffness of the structure is constantly updated.
- Currently a beam-like structure with through-the-width delaminations is considered. To approach a more realistic state of damage either the analysis should be subjected to several rotations or circular and rectangular plates should be considered, which are able of containing a delamination within the structure.
- A hybrid solution whereby parts of the analysis is carried out by FEA is conceivable and might yield the best compromise in terms of time vs. accuracy.

References

1. Suo, Z. and J.W. Hutchinson, *Interface crack between two elastic layers.* International Journal of Fracture, 1990. **43**(1): p. 1-18.
2. Olsson, R., *Mass criterion for wave controlled impact response of composite plates.* Composites Part A: Applied Science and Manufacturing, 2000. **31**(8): p. 879-887.
3. Schoeppner, G.A. and S. Abrate, *Delamination threshold loads for low velocity impact on composite laminates.* Composites Part A: Applied Science and Manufacturing, 2000. **31**(9): p. 903-915.
4. Kassapoglou, C., *Design and Analysis of Composite Structures: With Applications to Aerospace Structures*2011: John Wiley & Sons.
5. Orifici, A.C., I. Herszberg, and R.S. Thomson, *Review of methodologies for composite material modelling incorporating failure.* Composite Structures, 2008. **86**(1–3): p. 194-210.
6. Abrate, S., *Impact on Composite Structures*1998, New York: Cambridge University Press.
7. Dobyns, A.L., *Analysis of Simply-Supported Orthotropic Plates Subject to Static and Dynamic Loads.* AIAA Journal, 1981. **19**(5): p. 642-650.
8. Shivakumar, K.N., W. Elber, and W. Illg, *Prediction of Impact Force and Duration Due to Low-Velocity Impact on Circular Composite Laminates.* Journal of Applied Mechanics, 1985. **52**(3): p. 674-680.
9. Pang, S.S., et al., *Impact response of composite laminates with a hemispherical indenter.* Polymer Engineering & Science, 1991. **31**(20): p. 1461-1466.
10. Olsson, R., *Impact response of orthotropic composite plates predicted from a one-parameter differential equation.* AIAA, 1992. **30**(6): p. 1587-1596.
11. Swanson, S.R., *Limits of quasi-static solutions in impact of composite structures.* Composites Engineering, 1992. **2**(4): p. 261-267.
12. Reddy, J.N., *Mechanics of laminated composite plates and shells : theory and analysis*2004, Boca Raton: CRC Press.
13. Love, A.E.H., *The Stress Produced in a Semi-Infinite Solid by Pressure on Part of the Boundary.* Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1929. **228**(ArticleType: research-article / Full publication date: 1929 / Copyright © 1929 The Royal Society): p. 377-420.

14. Lekhnitskii, S.G., *Theory of Elasticity of an Anisotropic Elastic Body*1963: Holden-Day.

15. Dahan, M. and J. Zarka, *Elastic contact between a sphere and a semi infinite transversely isotropic body.* International Journal of Solids and Structures, 1977. **13**(3): p. 229-238.

16. Olsson, R. and S. Nilsson, *Simplified prediction of stresses in transversely isotropic composite plates under Hertzian contact load.* Composite Structures, 2006. **73**(1): p. 70-77.

17. Cairns, D.S., *Impact and post-impact response of graphite/epoxy and Kevlar/epoxy structures*, 1987.

18. Talagani, M.R., *A semi-analytical solution for the out-of-plane stresses for a quasi-static axi-symmetric indentation of a transversely isotropic plate using the Hankel transform.* Unpublished.

19. Hashin, Z., *Failure Criteria for Unidirectional Fiber Composites.* Journal of Applied Mechanics, 1980. **47**(2): p. 329-334.

20. Brewer, J.C. and P.A. Lagace, *Quadratic Stress Criterion for Initiation of Delamination.* Journal of Composite Materials, 1988. **22**(12): p. 1141-1155.

21. Hou, J.P., N. Petrinic, and C. Ruiz, *A delamination criterion for laminated composites under low-velocity impact.* Composites Science and Technology, 2001. **61**(14): p. 2069-2074.

22. Sun, C.T. and Z. Jin, *Fracture Mechanics*2011: Elsevier Science.

23. Rice, J.R., *A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks.* Journal of Applied Mechanics, 1968. **35**(2): p. 379-386.

24. Sih, G.C., *Handbook of Stress-intensity Factors for Researchers and Engineers: Stress-intensity Factor Solutions and Formulas for Reference*1974: Institute of Fracture and Solid Mechanics, Lehigh University.

25. Sun, C.T. and C.J. Jih, *On strain energy release rates for interfacial cracks in bi-material media.* Engineering Fracture Mechanics, 1987. **28**(1): p. 13-20.

26. Hwu, C. and J.S. Hu, *Stress intensity factors and energy release rates of delaminations in composite laminates.* Engineering Fracture Mechanics, 1992. **42**(6): p. 977-988.

27. Wang, J. and P. Qiao, *Interface crack between two shear deformable elastic layers.* Journal of the Mechanics and Physics of Solids, 2004. **52**(4): p. 891-905.

28. Andrews, M.G., *The static and dynamic interaction of multiple delaminations in plates subject to cylindrical bending*, in *Civil Engineering*2005, Northwestern University: Evanston. p. 218.

29. Suemasu, H. and O. Majima, *Multiple Delaminations and their Severity in Circular Axisymmetric Plates Subjected to Transverse Loading.* Journal of Composite Materials, 1996. **30**(4): p. 441-453.

30. Olsson, R., *Analytical prediction of large mass impact damage in composite laminates.* Composites Part A: Applied Science and Manufacturing, 2001. **32**(9): p. 1207-1215.

31. Olsson, R., *Closed form prediction of peak load and delamination onset under small mass impact.* Composite Structures, 2003. **59**(3): p. 341-349.

32. Olsson, R., M.V. Donadon, and B.G. Falzon, *Delamination threshold load for dynamic impact on plates.* International Journal of Solids and Structures, 2006. **43**(10): p. 3124-3141.

33. Zheng, S. and C.T. Sun, *DELAMINATION INTERACTION IN LAMINATED STRUCTURES.* Engineering Fracture Mechanics, 1998. **59**(2): p. 225-240.

34. Andrews, M.G., R. Massabó, and B.N. Cox, *Elastic interaction of multiple delaminations in plates subject to cylindrical bending.* International Journal of Solids and Structures, 2006. **43**(5): p. 855-886.

35. Liu, D., *Impact-Induced Delamination—A View of Bending Stiffness Mismatching.* Journal of Composite Materials, 1988. **22**(7): p. 674-692.

36. Choi, H.Y. and F.-K. Chang, *A Model for Predicting Damage in Graphite/Epoxy Laminated Composites Resulting from Low-Velocity Point Impact.* Journal of Composite Materials, 1992. **26**(14): p. 2134-2169.

37. Davies, G.A.O. and X. Zhang, *Impact damage prediction in carbon composite structures.* International Journal of Impact Engineering, 1995. **16**(1): p. 149-170.

38. Davies, G.A.O., D. Hitchings, and J. Ankersen, *Predicting delamination and debonding in modern aerospace composite structures.* Composites Science and Technology, 2006. **66**(6): p. 846-854.

39. Koloor, S.S.R., et al., *Evolution Characteristics of Delamination Damage in CFRP Composites Under Transverse Loading*, in *Damage and Fracture of Composite Materials and Structures*, M.N.N. Tamin, Editor 2012, Springer Berlin Heidelberg. p. 45-59.

40. Lopes, C.S., *Damage and Failure of Non-Conventional Composite Laminates*, in *Mechanics, Aerospace Structures & Materials*2009, Delft University of Technology: Delft.

41. (ASTM), A.S.f.T.a.M., *ASTM standard D7136 / D7136M, "Standard Test Method for Measuring the Damage Resistance of a Fiber-Reinforced Polymer Matrix Composite to a Drop-Weight Impact Event"*, in *Composite Standards*2005, ASTM International: West Conshohocken, PA.

42. Suemasu, H., S. Kerth, and M. Maier, *Indentation of Spherical Head Indentors on Transversely Isotropic Composite Plates.* Journal of Composite Materials, 1994. **28**(17): p. 1723-1739.

43. Leissa, A.W. and Y. Narita, *Vibration studies for simply supported symmetrically laminated rectangular plates.* Composite Structures, 1989. **12**(2): p. 113-132.

44. Matemilola, S.A. and W.J. Stronge, *Impact induced dynamic deformations and stresses in CFRP composite laminates.* Composites Engineering, 1995. **5**(2): p. 211-222.

45. Gdoutos, E.E., *Fracture mechanics : an introduction*2005, Dordrecht; Norwell, MA: Springer ; Distributed in North, Central and South America by Springer.

46. Krueger, R., P.J. Minguet, and K. O'Brien, *Implementation of interlaminar fracture mechanics in design: an overview*, in *14th International Conference on Composite Materials*2003: San Diego.

47. Dost, E.F., et al. *Effects of stacking sequence on impact damage resistance and residual strength for quasi-isotropic laminates*. 1991.

48. Hexcel. *HexPly 8551-7 Epoxy Matrix*. 2012 [cited 2012 27.11.2012]; Available from: http://hexcel.com/Resources/DataSheets/Prepreg-Data-Sheets/85517_us.pdf.

49. Craven, R., L. Iannucci, and R. Olsson, *Delamination buckling: A finite element study with realistic delamination shapes, multiple delaminations and fibre fracture cracks.* Composites Part A: Applied Science and Manufacturing, 2010. **41**(5): p. 684-692.

Appendix A – MATLAB Code

```
%% -------------------------- Master_simV2.m ------------------------------------ %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% masterfile for the simulation of a low-mass, low-velocity impact event on a given
% structure. A quasi-static approach is used (justified by the paper from
% Swanson(1991), which will have an influence on the way the impact force is
% calculated (see Pang et al, 1991) but also how the simulation is carried out (see
% Weirdie and Lagace, 1998)
% The flowcharrt from notebook 1, p.112 will be followed in order to arrive at the
% final damage state in the structure. To this end, a variety of modules will be
% called in order to set this up as flexible as possible.
%-------------------------------------------------------------------------------%
%  the various modules are:
% - input, setting up the problem, material used, geometry of the structure, boundary
% conditions applied
%
% - impact_load, computes the impact load and indentation during the impact
%
% - stress_state_pristine, calculates the stresses in the structure until the first
% failure is detected, based upon a selected failure criterion
%
% - delamination_tool, module for the computation of the beam segement displacement,
% energy release rate and the virtual crack growth, (stress concentration factors),
% updates the stiffness of the structure for subsequent stress calculations
%
% - stress_state_damaged, once the structure is damaged, it's internal stress state
% will change
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V2
% date: 15.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% MATLAB initiation
clear all
close all
clc
format short e

%% Variable declaration
% the global variable "dam" will be used to pass information between the LEFM_tool
% and the master file
global dam tar imp sim

%% Input
% load the input file
input_funcV2();

% create filename
filename = ['LAM' num2str(sim.lam) '_' num2str(imp.Ei) 'J_C'...
   num2str(sim.contact) '.mat'];
fprintf(1,['Simulation data will be saved to "' filename '" ...\n'])

%% Obtain impact histories
% impact load Pt, indentation alpha and central deflection wc
[impact] = impact_funcV4();
Pt = impact(:,1);
```

```
alpha = impact(:,2);
wc = impact(:,3);

% save the displacement to structure SIM
SIM.wa = wc';
SIM.tvec = impact(:,4);

%% 1st leg - until first damage
% call the stress computation programme for the undamaged laminate. Indicate with the
% flag which contact stress computation is to be used: "0" - simple analytical version,
% based on Dahan and Zarka or "1" - Cairns solution. Increase the load until the
% first "real" delamination is found.
stressflag = 0;

% matrix FAIL indicates at which interface and at which distance away from the impact
% the first "cell" fails. the vector failinfo summarizes that by stating through the
% thickness position (2) and position away from the center in [m](1).

pl = 1;   % start index for the load vector
failflag = 0;

SIM.P(1) = 0;
while failflag == 0 && pl <= size(impact,1)
   pl = pl + 1;
   Papp(pl) = tar.K_b*wc(pl);
   fprintf(1, 'current load level: %1.4e N at % 1.4e m displacement\n',...
      Papp(pl), wc(pl))

%      keyboard
   % compute the stress-state in the undamaged laminate and analyse the beam for
   % failure
   [SIGPLY] = stress_state_pristine_funcV3(Papp(pl), stressflag);
   [failflag, failinfo, FAIL] = fail_analysisV2(SIGPLY, alpha(pl));

   % save Papp
   SIM.P(pl) = Papp(pl);
end


%% 2nd leg - until the end of impact
fprintf(1, 'switching from 1st to 2nd leg\n')

% prep "failinfo" for LEFM_tool
dam.delams = [failinfo(:,1) failinfo(:,2) zeros(size(failinfo,1),1)];

while pl <= size(impact,1)
   tempdelams = dam.delams;

   % because now damage has been initialized an updated force has to be determined
   Papp(pl) = Pupdate_funcV1(wc(pl), sim.contact);

   fprintf(1, '\ncurrent load level: %1.4e N at %1.4e m displacement\n',...
      Papp(pl), wc(pl))

   %------------------------ call LEFM_tool ---------------------------------------%
   [Papp(pl)] = LEFM_toolV3(Papp(pl), wc(pl), sim.contact);

    % save G, SIF, G_k, and length to a structure
```

```
[~,temp] = sort(dam.delams(:,2), 'ascend');
for dl = 1:size(dam.delams,1)
   intnum = round(dam.delams(temp(dl),2)/tar.E0(7));
   SIM.at{intnum}(pl) = dam.delams(temp(dl),1);
   SIM.Gt{intnum}(pl) = dam.G(temp(dl));
   SIM.SIFt{intnum}(1:2,pl) = dam.SIF(temp(dl), 1:2);
   SIM.Gkt{intnum}(pl) = dam.SIF(temp(dl),5);
end

% check if any of the delaminations has reached the boundary of the beam
if max(dam.delams(:,1)) >= tar.L/2
   fprintf(1, 'laminate failed prematurely!!\n')
   break
end

   %% Stress Update
% compute the new stress-state and analyse for failure
[SIGPLY] = stress_state_damaged_funcV1(Papp(pl), stressflag);
[failflag, failinfo, FAIL] = fail_analysisV2(SIGPLY, alpha(pl));

   %% growth update
% if we have growth than the newly grown delamination should overwrite the
% failinfo line.

for dl = 1:size(failinfo,1)
  % check whether dam.delams needs updating. Either it is not growing, or the
  % failure predicition has progressed in terms of radial value
  % first find the corresponding line in failinfo
  temp1 = find(dam.delams(:,2) == failinfo(dl,2));
  if isempty(temp1)
    % new entry for dam.delams
    fprintf(1,'new delamination at interface %i is added to "delams".\n',...
      round(failinfo(dl,2)/tar.E0(7)))
    dam.delams(end+1,:) = [failinfo(dl,1) failinfo(dl,2) failinfo(dl,3)];
  else
    % this interface already exists in delams. Choose whether it should be
    % "updated" in terms of radial distance (update = 1) or left until it
    % will grow (update = 0).

    switch sim.update
      case 0
        fprintf(1, 'no update!\n')

      case 1

        % Either it is not growing, or the failure predicition has
        % progressed in terms of radial value
        if dam.delams(temp1,3)==0 && failinfo(dl,1)<=dam.delams(temp1,1)
          fprintf(1,'no changes to "delams".\n')
        elseif dam.delams(temp1,3) == 1;
          fprintf(1,'delamination is growing. no input from failinfo.\n')
        else
          fprintf(1,...
            '"delams" will be adjusted with information from failinfo.\n')
          dam.delams(temp1,1) = failinfo(dl,1);
```

```
            end
        end
    end
  end
%     keyboard

  % save Papp
  SIM.P(pl) = Papp(pl);

  % update loop variables
  pl = pl + 1;

  % save SIM structure to mat file
  save(filename, '-struct', 'SIM')

end

if wc(pl-1) == max(wc)
  fprintf(1, 'max displacment of %1.4e m reached. Simulation stopped. \n', max(wc))
end


function [K,EngCst,As] = ABDmat(E0,seq, orient, flag)
% E0 - the ply properties of the system
% seq - stacking sequence in degrees
% orient - reference orientation in axisymmetric problems
% flag - switch between cartesian(0) and axisymmetyric(1)
% returns K: A,B,D and Dred matrix
% returns EngCst: Engineering constants taken from Christos' book

%% compute the z-vector
z = zvector(E0(7), length(seq));
h = length(seq)*E0(7);
%% write stiffness matrix for orthotropic plane stress layer
Q_xx = E0(1) / (1 - E0(4)^2 * E0(2)/E0(1));
Q_yy = E0(2) / (1 - E0(4)^2 * E0(2)/E0(1));
Q_xy = E0(4) * E0(2) / (1 - E0(4)^2 * E0(2)/E0(1));
Q_ss = E0(3);
Q_ss1 = E0(3); % unless I get other values for G23, G13
Q_ss2 = E0(3);

%% Determining of the ABD matrices

% depending on which approach has been used...
switch flag
  case 0
  % case 0 represents the standard cartesian version taken from Christos;
  % book

    % initialize ABD matrix
    As = zeros(2,2);
    A = zeros(3,3);
    B = zeros(3,3);
    D = zeros(3,3);

    for k = 1:length(seq)

      m = cosd(seq(k));
```

```
    n = sind(seq(k));

    %transformed Stiffness matrix, approach from Kassapoglou's book
    Q(1,1) = (m^4 * Q_xx) + (n^4 * Q_yy) + (2 * m^2 * n^2 * Q_xy) +...
        (4 * m^2 * n^2 * Q_ss);
    Q(2,2) = (n^4 * Q_xx) + (m^4 * Q_yy) + (2 * m^2 * n^2 * Q_xy) +...
        (4 * m^2 * n^2 * Q_ss);
    Q(1,2) = (m^2 * n^2 * Q_xx) + (m^2 * n^2 * Q_yy) + (m^4 + n^4) * Q_xy...
        - (4 * m^2 * n^2 * Q_ss);
    Q(3,3) = (m^2 * n^2 * Q_xx) + (m^2 * n^2 * Q_yy) -...
        (2 * m^2 * n^2 * Q_xy) + (m^2 - n^2)^2 * Q_ss;
    Q(1,3) = (m^3 * n * Q_xx) - (m * n^3 * Q_yy) + (m*n^3 - m^3*n) *...
        Q_xy + 2 * (m*n^3 - m^3*n) * Q_ss;
    Q(2,3) = (m * n^3 * Q_xx) - (m^3 * n * Q_yy) + (m^3*n - m*n^3) *...
        Q_xy + 2 * (m^3*n - m*n^3) * Q_ss;
    Q(2,1) = Q(1,2);
    Q(3,1) = Q(1,3);
    Q(3,2) = Q(2,3);
    % transverse shear stiffnesses,
    Q(4,4) = Q_ss1*m^2 + Q_ss2*n^2;
    Q(4,5) = (Q_ss1-Q_ss2)*m*n;
    Q(5,5) = Q_ss1*n^2 + Q_ss2*m^2;
    Q(5,4) = Q(4,5);

    % writing the ABD matrix
    for i = 1:3
        for j = 1:3
            A(i,j) = A(i,j) + Q(i,j)*(z(k+1) - z(k));
            B(i,j) = B(i,j) + 0.5 * Q(i,j)*(z(k+1)^2 - z(k)^2);
            D(i,j) = D(i,j) + 1/3 * Q(i,j)*(z(k+1)^3 - z(k)^3);
        end
    end
    for i = 4:5
        for j = 4:5
            As(i-3,j-3) = As(i-3,j-3) + Q(i,j)*(z(k+1) - z(k));
        end
    end
end

% collecting everything
Dred = D - B*inv(A)*B;
K = [A, B; D, Dred];

% only for symmetric layups
alpha = inv(A);
beta = zeros(3);
delta = inv(D);

% Engineering constants
EngCst(1,1) = 1 / (h * alpha(1,1));      %E1m
EngCst(1,2) = 1 / (h * alpha(2,2));      %E2m
EngCst(2,1) = 12 / (h^3 * delta(1,1));    %E1b
EngCst(2,2) = 12 / (h^3 * delta(2,2));    %E2b
EngCst(3,1) = 1 / (h * alpha(3,3));      %G12m
```

```
    EngCst(3,2) = 12 / (h^3 * delta(3,3));     %G12b
    EngCst(4,1) = -alpha(1,2) / alpha(2,2);    %nu12m
    EngCst(4,2) = -alpha(1,2) / alpha(1,1);    %nu21m
    EngCst(5,1) = -delta(1,2) / delta(1,1);    %nu12b
    EngCst(5,2) = -delta(1,2) / delta(2,2);    %nu21b


  case 1
  % case 1 uses the invariants and is used here for the axisymmetric
  % approach

    % initialize ABD matrix
    A = zeros(2,2);
    B = zeros(2,2);
    D = zeros(2,2);

    % Determining the invariants
    U(1) = (3*Q_xx + 3*Q_yy + 2*Q_xy + 4*Q_ss)/8;
    U(2) = (Q_xx - Q_yy)/2;
    U(3) = (Q_xx + Q_yy - 2*Q_xy - 4*Q_ss)/8;
    U(4) = (Q_xx + Q_yy + 6*Q_xy - 4*Q_ss)/8;
    U(5) = (Q_xx + Q_yy - 2*Q_xy + 4*Q_ss)/8;

    for k = 1:length(seq)
      % determine the correct angle theta
      if orient > seq(k)
        theta = orient-seq(k);
      else
        theta = seq(k)-orient;
      end

      m1 = cosd(2*theta);
      m2 = cosd(4*theta);
      n1 = sind(2*theta);
      n2 = sind(4*theta);

      Q(1,1) = U(1) + U(2)*m1 + U(3)*m2;
      Q(1,2) = U(4) - U(3)*m2;
      Q(2,2) = U(1) - U(2)*m1 + U(3)*m2;
      Q(2,1) = Q(1,2);

      for i = 1:2
        for j = 1:2
          A(i,j) = A(i,j) + Q(i,j)*(z(k+1) - z(k));
          B(i,j) = B(i,j) + 0.5 * Q(i,j)*(z(k+1)^2 - z(k)^2);
          D(i,j) = D(i,j) + 1/3 * Q(i,j)*(z(k+1)^3 - z(k)^3);
        end
      end
    end

    % collecting everything
    Dred = D - B*inv(A)*B;
    K = [A, B; D, Dred];
    EngCst = [];

end
```

```
end

function [z] = zvector(t_ply, ne)

  if mod(ne,2) == 0      %check for even number of plies
  %     display('even')
    for j = 1:ne+1
      if (j <(ne+1)/2)
        z(j,1) = -(ne/2-j+1)* t_ply;
      end
      if (j==(ne+1)/2)
        z(j,1)=0;
      end
      if (j >(ne+1)/2)
        z(j,1) = -(ne/2-j+1)* t_ply;
      end
    end
  else
  %     display('odd')
    for j = 1:ne+1
      if (j <=(ne+1)/2)
        z(j,1) = (-((ne+1)/2-j+1) + 0.5)* t_ply;
      end

      if (j >=(ne+1)/2)
        z(j,1) = -((ne+1)/2-j+1 - 0.5)* t_ply;
      end
    end
  end
end


%% ------------------------- input_funcV2.m ---------------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This fucntion serves as input file for the simulation of the impact. The user has
% to give as input the following items:
%
% impactor:
% - radius of the sphere, imp.R [m]
% - Young's modulus, imp.E [N/m^2]
% - Poisson ratio, imp.nu [-]
% - mass, imp.M [kg]
% - velocity, imp.v0 [m/sec]
%
% target:
% - density of cured material, tar.rho [kg/m^3]
% - basic ply properties, tar.E0 [1x7] whereby:
%     * E0(1-3): Young's modulus in fibre and transverse direction and shear modulus,
%     respectively [N/m^2]
%     * E0(4-6): Poisson ratio in 12, 21, and 13 direction, respectively [-]
%     * E0(7): cured ply thickness [m]
% - critical fracture toughness, tar.Gcr [Nm^(-3/2)]
% - beam length, tar.L [m]
% - beam width, tar.w [m]
% - stacking sequence, lam_input
% - boundary conditions, tar.BC (0 -> simply supported, 1 -> clamped)
```

```
%
% Additionally, the following parameters are being computed
% - transversely isotropic stiffnesses, tar.Etransiso [1x6] whereby:
%     * Etransiso(1): Young's modulus in r-direction [N/m^2]
%     * Etransiso(2): Poisson ration in r_theta-direction [-]
%     * Etransiso(3): shear modulus in r_theta-direction [N/m^2]
%     * Etransiso(4): Young's modulus in z-direction [N/m^2]
%     * Etransiso(5): Poisson ratio in rz-direction [N/m^2]
%     * Etransiso(6): shear modulus in rz-direction [N/m^2]
% - heigh of the beam, tar.h [m]
% - mass of the beam, tar.M [kg]
%
% The two structures "imp" and "tar" are made available to the rest of the programme
% by declaring them globally.
%-------------------------------------------------------------------------------%
% Author: J.J. Kurpierz
% Version: V2
% Date: 04.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% start of function
function [] = input_funcV2()

%% global variable declaration
global imp tar sim

%% Input Parameters
%---------------------------- Simulation input ----------------------------------%
% input for the way fail_analysis is running
% break (=1) or don't (=0) when first failure is detected
sim.br = 0;
% how to incorporate singular stress field: 1: deepest embedded, 2: longest, 3: all
sim.SIF = 3;
% update (=1) dam.delams or not (=0)
sim.update = 1;
% contact: 0: unconstrained, 1: constrained
sim.contact = 0;
%---------------------------- Impactor ------------------------------------------%
% stainless hardened steel 17-4 PH, H900, taken from Martin's experiment
imp.R = 0.008;                % impactor Radius, [m]
imp.E = 197*1e9;               % impactor E-modulus [N/m^2]
imp.nu = 0.272;               % impactor Poisson ratio [-]
imp.M = 5.44;                 % impactor mass [kg]
imp.Ei = 15;                 % incident impact energy
imp.v0 = sqrt(2*imp.Ei/imp.M);        % impactor velocity [m/s]

%---------------------------- target --------------------------------------------%
tar.rho = 1590;               % plate density, [kg/m^3]

% properties for IM7/8551-7 taken from HexPly, complemented where necessary with
% AS4 data from Lopes
tar.E0(1)=144.8*1e9;             % E-mod in fibre direction [Pa]
tar.E0(2)=8.3*1e9;              % E-mod in transverse direction [Pa]
tar.E0(3)=5.8*1e9;              % inplane shear mod [Pa]
tar.E0(4)=0.32;               % poisson ratio 12 [-], nu_lt
tar.E0(5)=tar.E0(4)*tar.E0(2)/tar.E0(1);    % poisson ratio in 21, nu_tl
tar.E0(6)=0.3487;              % poisson ratio in 13, nu_tt
tar.E0(7)=0.19E-3;              % ply thickness

% fracture toughness for GII mode for IM7/8552 (O'Brien), [Nm^(-3/2)]
tar.Gcr = 1138;
```

```
% layup input. If the last digit is equal to "1", then the laminate is mirrored.
% Otherise a "0" has to be placed at the last entry.
LAM{1} = [45 90 -45 0 45 90 -45 0 45 90 -45 0 1];
LAM{2} = [45 45 45 90 90 90 -45 -45 -45 0 0 0 1];
LAM{3} = [30 60 90 -60 -30 0 30 60 90 -60 -30 0 1];
LAM{4} = [30 60 90 -30 -60 0 30 60 90 -30 -60 0 1];
LAM{5} = [45 90 -45 90 -45 90 -45 0 45 0 45 0 1];
LAM{6} = [45 0 -45 0 -45 0 -45 90 45 90 45 90 1];
LAM{7} = [45 45 90 90 -45 -45 0 0 45 45 90 90 -45 -45 0 0 1];
LAM{8} = [-45 0 45 90 -45 0 45 90 -45 0 45 90 1];
LAM{9} = [-45 -45 0 0 45 45 90 90 -45 -45 0 0 45 45 90 90 1];
% choose the laminate to be analyzed
sim.lam = 9;

lam_input = LAM{sim.lam};

if lam_input(end) == 1
   tar.seq = [lam_input(1:end-1) lam_input(end-1:-1:1)];
else
   tar.seq = lam_input(1:end-1);
end

% geometry
% tar.L = 0.127;                                          % beam length, [m]
% tar.w = 0.0762;                                          % beam width, [m]

% rotated geometry
tar.L = 0.0762;                      % beam length, [m]
tar.w = 0.127;                      % beam width, [m]

% boundary condtions
tar.BC = 0;

%% computed values
tar.h = length(tar.seq)*tar.E0(7);           % plate thickness, [m]
tar.M = tar.rho*tar.L*tar.h*tar.w;            % beam mass, [m]
% Engineering Constants according to Kasapoglou
[~, EngCst] = ABDmat(tar.E0, tar.seq, 0, 0);

% according to Abrate
tar.Etransiso(1) = (EngCst(1,1)+EngCst(1,2))/2;  % E-mod in radial direction [N/m^2]
tar.Etransiso(2) = (EngCst(4,1)+EngCst(4,2))/2;  % in plane Poisson ratio [-]
tar.Etransiso(3) = EngCst(3,1);

% Engineering Constants for axisymmetric problems according to Suemasu
del = 1 - 2*tar.E0(5)*tar.E0(4)*tar.E0(5) - 2*tar.E0(6)*tar.E0(4) - tar.E0(6)^2;
A11 = tar.E0(1)*(1-tar.E0(6)^2)/del;
A22 = tar.E0(2)*(1-tar.E0(4)*tar.E0(5))/del;
A12 = tar.E0(2)*tar.E0(4)*(1+tar.E0(6))/del;
A23 = tar.E0(2)*(tar.E0(6)+tar.E0(4)*tar.E0(5))/del;

% E-mod in z-direction [N/m^2]
tar.Etransiso(4) =  A22 - (A12 + A23)^2/(A11+A22+2*A12);
% out-of-plane Poisson ratio [-]
```

```
tar.Etransiso(5) = (A12 + A23)/(A11+A22+2*A12);
% shear mod in z direction [N/m^2]
tar.Etransiso(6) = 2*(1/tar.E0(3)+1/((1-tar.E0(6))*tar.E0(3)))^(-1);

end


%% ------------------------- impact_load_funcV3.m ------------------------------ %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This module determines the load history from the impact event for an undamaged
% laminate and returns the laod and indentation vector to the master file.
% It is based on the paper from Pang et al (1991) and is valid for thin plates, small
% deflection, i.e. Ks = 0, Km = 0.
%-------------------------------------------------------------------------------%
% Input:
% no input from outside is neccessary as long as the two structures "imp" and "tar"
% are defined globally previously.
%-------------------------------------------------------------------------------%
% Output:
% - matrix "impact" (dt,2)
%     * impact(:,1) = impact load P(t)
%     * impact(:,2) = indendation alpha(t)
%-------------------------------------------------------------------------------%
% Author: J.J. Kurpierz
% Version: V1
% Date: 13.07.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function [impact] = impact_funcV4()

%% variable decleration
global imp tar

Er = tar.Etransiso(1);
Ez = tar.Etransiso(4);
Grz = tar.Etransiso(6);
nurth = tar.Etransiso(2);
nurz = tar.Etransiso(5);

%% Preliminary calculations
%-------------------------------------------------------------------------------%
% compute contact stiffness according to Greszczuk
del = 1-nurth - 2*nurz^2;
Crr = Er*(1-nurz^2)/((1+nurth)*del);
Czz = Ez*(1-nurth)/del;
Crz = Er*nurz/del;

% anisotropic (target) part
tar.K = sqrt(Crr)*((sqrt(Czz*Crr)+Grz)^2-(Crz+Grz)^2)^0.5 /...
   (2*pi*sqrt(Grz)*(Czz*Crr-Crz^2));

% isotropic impactor part
imp.K = (1-imp.nu^2)/(pi*imp.E);

% contact stiffness parameter
K_con = 4*sqrt(imp.R) / (3*pi*(imp.K+tar.K));

%-------------------------------------------------------------------------------%
% equivalent mass, based on Swanson(1992),
```

```
if tar.BC == 0
   m_eq = 0.486 * tar.M;
else
   m_eq = 0.371 * tar.M;
end

% equivalent stiffness based on laminated beam theory
[~, EngCst, ~] = ABDmat(tar.E0, tar.seq, 0, 0);
Eb = EngCst(2,1); % using E1_b
% moment of inertia of the beam
I = 1/12*tar.h^3*tar.w;
% equivalent stiffness for either simply supported or clamped beam
if tar.BC == 0
   tar.K_b = 48*Eb*I/tar.L^3;
else
   tar.K_b = 192*Eb*I/tar.L^3;
end
K_eq = tar.K_b;

%% iterative solution of equation 3.3.11
% initial value for the indendation [m]
alpha_init = 0;
options = optimset('Display', 'off');
alpha_m = fsolve(@(x) imp.v0^2*imp.M - K_con * x^(5/2) - K_con^2/K_eq*...
               (1+m_eq/imp.M) * x^3 + 2/3*K_con^3*m_eq/...
               (imp.v0^2*imp.M^2*K_eq) * x^(11/2), alpha_init, options);

%% computation of impact histories
% max force during impact [N] based on Hertzian contact theory
P_m = K_con * alpha_m^(3/2);
% time duration [sec] to reach max force
t_m = imp.v0*pi*imp.M / (2*P_m);
% duration of the impact
dt = 500;
t_vec = 0:t_m/dt:t_m;
% force, indendation and displacment history
P_t = P_m * sin(pi/(2*t_m)*t_vec);
alpha_t = alpha_m * sin(pi/(2*t_m)*t_vec).^(2/3);
wc_t = 4*P_m*t_m^2/(pi^2*imp.M)*sin(pi/(2*t_m)*t_vec)+...
   (imp.v0-2*P_m*t_m/(pi*imp.M))*t_vec - alpha_m*sin(pi/(2*t_m)*t_vec).^(2/3);
%% function output
impact = [P_t' , alpha_t', wc_t', (t_vec/t_m)'];

end


%% ------------------------- stress_state_pristine_funcV3.m --------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This module determines the stress state for a given impact load input. To this end
% it computes the stresses in the entire laminate due to bending and subsequently
% computes the transverse stresses resulting from the bending.
% Furthermore the contact stresses due to impact are computed by one of the various
% solution techniques available.
% For the bending solution the laminated beam theory has been used given in Reddy.
%-------------------------------------------------------------------------------%
% Input:
```

```
% - current load level, Pt
% - stressflag, indicates whether (1) or not (0) the finite-thickness approach is
% used
%-------------------------------------------------------------------------------%
% Output:
% - FAIL, matrix indicating at each interface for every step in radial direction
% whether or not the material failed (0, 1). To be used as input for the delamination
% module
%-------------------------------------------------------------------------------%
% Author: J.J. Kurpierz
% Date: 05.11.2012
% Version: V3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of the function
function [SIGPLY] = stress_state_pristine_funcV3(Pt, stressflag)

%% declare variables
global tar resx

%% initial setup
% resolution in x direction (equivalent to r)
resx = round(5*tar.L/tar.E0(7));

% resolution in z direction ( a value of 1 means that only the interfaces are being
% evaluated)
resz = 1;

%% stress computation
%---------------------------- flexural stresses ---------------------------------%
% D-matrix from the input file
[K,~] = ABDmat(tar.E0, tar.seq, 0, 0);
D = K(4:6,1:3);
d = inv(D);
if tar.BC == 0
   Mr = 0;
else
   Mr = -1/8*Pt*tar.L;
end
SIGB = flex_sigV2(Pt, d, Mr, resx, resz);

%---------------------------- contact stresses ----------------------------------%
SIGC = contact_sigV1(Pt, stressflag, resx, resz);

SIGPLY.B = SIGB;
SIGPLY.C = SIGC;

%---------------------------- transformation to ply coordinates -----------------%
% The global stress components can now be added together and then transformed to
% yield the stress components in the ply coordinate system. From the bending stress
% analysis we get two values per interface. So the ply based matrices should also
% have two values per interface, which then will be passed to the failure criterion.
% Add flexural and contact stresses (at the same time duplicate the transverse stress
% components)

for rl = 1:resx
   for k = 1:length(tar.seq)-1
      % define global stress vector
      % upper ply first
      SIGglobu = [SIGB.XX(2*(k-1)+1,rl) + SIGC.XX(k,rl);
```

```
            SIGB.YY(2*(k-1)+1,rl) + SIGC.YY(k,rl);
            SIGC.ZZ(k,rl);
            SIGB.XZ(k,rl) + SIGC.XZ(k,rl);
            SIGB.YZ(k,rl);
            SIGB.XY(2*(k-1)+1,rl)];

    % lower ply first
    SIGglobl = [SIGB.XX(2*(k-1)+2,rl) + SIGC.XX(k,rl);
            SIGB.YY(2*(k-1)+2,rl) + SIGC.YY(k,rl);
            SIGC.ZZ(k,rl);
            SIGB.XZ(k,rl) + SIGC.XZ(k,rl);
            SIGB.YZ(k,rl);
            SIGB.XY(2*(k-1)+2,rl)];

    % apply transformation
    % transformation matrix (see equation 3.4.10)
    mu = cosd(tar.seq(k));
    nu = sind(tar.seq(k));
    TRANSU = [mu^2 nu^2 0 0 0 2*mu*nu;
          nu^2 mu^2 0 0 0 -2*mu*nu;
          0 0 1 0 0 0;
          0 0 0 mu -nu 0;
          0 0 0 nu mu 0;
          -nu*mu mu*nu 0 0 0 mu^2-nu^2];
    SIGlocu = TRANSU*SIGglobu;
    ml = cosd(tar.seq(k+1));
    nl = sind(tar.seq(k+1));
    TRANSL = [ml^2 nl^2 0 0 0 2*ml*nl;
          nl^2 ml^2 0 0 0 -2*ml*nl;
          0 0 1 0 0 0;
          0 0 0 ml -nl 0;
          0 0 0 nl ml 0;
          -nl*ml ml*nl 0 0 0 ml^2-nl^2];
    SIGlocl = TRANSL*SIGglobl;

    % reorganize into a structure with 4 matrices to be used by Failure criterion
    SIGPLY.S1(2*(k-1)+1, rl) = SIGlocu(1);
    SIGPLY.S3(2*(k-1)+1, rl) = SIGlocu(3);
    SIGPLY.S4(2*(k-1)+1, rl) = SIGlocu(4);
    SIGPLY.S5(2*(k-1)+1, rl) = SIGlocu(5);
    SIGPLY.S1(2*(k-1)+2, rl) = SIGlocl(1);
    SIGPLY.S3(2*(k-1)+2, rl) = SIGlocl(3);
    SIGPLY.S4(2*(k-1)+2, rl) = SIGlocl(4);
    SIGPLY.S5(2*(k-1)+2, rl) = SIGlocl(5);

  end
end

end

%% ------------------------- contact_sigV1.m --------------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This module computes the contact stresses in the impacted beam. There are two
% possibilities. Either a fast assessment is desired in which case only the full
```

```
% analytical approach by Dahan&Zarka can be used. Otherwise the full solution based
% on Cairns thesis will be used resulting in longer computation times. The variable
% "stressflag" is used to activate the use of the latter.
%-------------------------------------------------------------------------------%
% input:
% - current load level, Pt [N]
% - stressflag to indicate whether Cairns solution is computed as well [-]
% - resolution in x direction, resx [-]
% - resolution in z direction, resz [-] (if resz = 1 then we only look at the
%-------------------------------------------------------------------------------%
% output:
% - structure SIGC containing 4 fields
%     * .xx (ni,resx)
%     * .yy (ni,resx)
%     * .zz (ni, resx)
%     * .xz (ni, resx)
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% version: V1
% date 05.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function[SIGC] = contact_sigV1(Pt, stressflag, resx, resz)

%% variable declaration
global tar imp
P = Pt;
Er = tar.Etransiso(1);
Ez = tar.Etransiso(4);
Grz = tar.Etransiso(6);
nurth = tar.Etransiso(2);
nurz = tar.Etransiso(5);

% Preliminaries taken from Lekhnitskii
% half-space compliance matrix
a11 = 1/Er;
a12 = -nurth/Er;
a13 = -nurz/Ez;
a33 = 1/Ez;
a44 = 1/Grz;

% constants defined to simplify the expressions
del = a11*a33-a13^2;
a = a13*(a11-a12)/del;
b = (a13*(a13+a44)-a12*a33)/del;
c = (a13*(a11-a12) + a11*a44)/del;
d = (a11^2 - a12^2)/del;

s1 = sqrt((a+c+sqrt((a+c)^2-4*d))/(2*d));
s2 = sqrt((a+c-sqrt((a+c)^2-4*d))/(2*d));

% contact radius and pressure from Hertzian loading
Rc = (3*pi/4*abs(P)*imp.R*(imp.K+tar.K))^(1/3);
p0 = 3*P/(2*pi*Rc^2);

%% switch between the different possibilities
switch stressflag
```

```
case 0 % closed form solution, taken from Dahan and Zarka
  % some definitions used later
  p1 = 1-a*s1^2;
  p2 = 1-a*s2^2;
  q1 = (b-a*s2^2)*p1;
  q2 = (b-a*s1^2)*p2;
  nu = (b-1)*sqrt(d)/(a*c-d);
  mu = (b-1)*(a+sqrt(d))/(a*c-d);
  k1 = s1^2/((s2-s1)*sqrt(d));
  k2 = s2^2/((s2-s1)*sqrt(d));
  l1 = nu*s1^2*p2/(s2-s1);
  l2 = nu*s2^2*p1/(s2-s1);
  % Rp is the "circular" plate which is affected locally
  Rp = 15*Rc;
  z = 0:tar.E0(7)/resz:tar.h;

  % initialization
  SIG_RR = zeros(length(z)-2,resx);
  SIG_TT = zeros(length(z)-2,resx);
  SIG_ZZ = zeros(length(z)-2,resx);
  SIG_RZ = zeros(length(z)-2,resx);

  r = 0;
  rl = 1;
  while r < Rp
    % the loop over the thickness starts at the first interface and stops at the
    % last
    for zl = 2:length(z)-1
      S(1) = s1;
      S(2) = s2;
      for k = 1:2
        %gamma^4
        gamma(k) = 1/Rc^4*(r^2-Rc^2+S(k)^2*z(zl)^2)^2 ...
          + 4*S(k)^2*z(zl)^2/Rc^2;
        %alpha^2
        alpha(k) = 1/2*(r^2+S(k)^2*z(zl)^2-Rc^2+...
              Rc^2*sqrt(gamma(k)));
        %beta
        beta(k) = -S(k)*z(zl)*Rc/sqrt(alpha(k));
        % remaining quantities for the computation
        S1(k) = atan((Rc-beta(k))/(sqrt(alpha(k))+S(k)*z(zl)));
        S1p(k) = (beta(k)+Rc)/r;
        T1p(k) = (sqrt(alpha(k))-S(k)*z(zl))/r;
        D2(k) = 1/(2*Rc)*(r*S1(k)-S(k)*z(zl)*S1p(k)-Rc*T1p(k));
        C2(k) = 1-1/Rc*(r*S1p(k) + S(k)*z(zl)*S1(k));
        D3(k) = 1/(3*Rc)*(Rc^2-r^2+S(k)^2*z(zl)^2/2)*S1p(k)+...
          S(k)*z(zl)/6*T1p(k)-r*S(k)*z(zl)/(2*Rc)*S1(k)+r/3;
      end

      if rl == 1
        SIG_RZ(zl-1,rl) = 0;
        SIG_ZZ(zl-1,rl) = -p0*(1 + S(1)*S(2)/(S(1)-S(2))*z(zl)/Rc*...
                (atan(Rc/(S(1)*z(zl)))...
                -atan(Rc/(S(2)*z(zl)))));
```

```
            SIG_RR(zl-1,rl) = p0*(mu/2-1/sqrt(d)-z(zl)/Rc*((k1-l1/2)*...
               atan(Rc/(S(1)*z(zl)))-(k2-l2/2)*atan(Rc/(S(2)*z(zl)))));
            SIG_TT(zl-1,rl) =  SIG_RR(zl-1,rl);
          else
            SIG_RZ(zl-1,rl) = p0*(D2(1)-D2(2))/((s1-s2)*sqrt(d));
            SIG_ZZ(zl-1,rl) = p0*(S(2)*C2(1)-S(1)*C2(2))/(S(1)-S(2));
            SIG_RR(zl-1,rl) = p0*(-1/((S(1)-S(2))*sqrt(d))*(S(1)*C2(1)-...
               S(2)*C2(2))+nu/(r*(S(1)-S(2)))*(S(1)*p2*D3(1)-S(2)*p1*D3(2)));
            SIG_TT(zl-1,rl) = p0*(sqrt(d)/((S(1)-S(2))*(a*c-d))*(S(1)*q1*C2(1)...
              -S(2)*q2*C2(2))-nu/(r*(S(1)-S(2)))*(S(1)*p2*D3(1)-S(2)*p1*D3(2)));
          end
        end
        % loop variable update
        rl = rl + 1;
        r = r+tar.L/2/resx;
      end

      SIGC.XX = SIG_RR;
      SIGC.YY = SIG_TT;
      SIGC.XZ = SIG_RZ;
      SIGC.ZZ = SIG_ZZ;

    case 1 % solution taken from Cairns

      % computation set up
      Rp = 20*Rc;
      z = -tar.h/2:tar.E0(7)/resz:tar.h/2;
    %     % computation location for r
    %     dr = 5;                     % discretization in r direction within Rp
    %     r = 0:Rc/dr:Rp;

      % initialization
      SIG_RR = zeros(length(z)-1,resx);
      SIG_TT = zeros(length(z)-1,resx);
      SIG_ZZ = zeros(length(z)-1,resx);
      SIG_RZ = zeros(length(z)-1,resx);

      % conditions at the impact surface
      ra = 0;
      ral = 1;
      while ra <= Rc
        SIG_ZZ_b(ral) = p0/Rc*(Rc^2-ra^2).^(1/2);
        ral = ral + 1;
        ra = ra + 0.5*tar.L/resx;
      end
      SIG_ZZ_b(ral:resx) = SIG_ZZ(ral:resx);

      % max number of harmonics
      kmax = 150;
      r = 0;
      rl = 1;
      while r < Rp

        for zi = 1:length(z)-1
          k = 1;
```

```
while SIG_ZZ(1,rl) <= SIG_ZZ_b(1,rl) && k <= kmax

  % compute the roots of the zeroth order bessel function
  mue = zerobess('J', 0,k);
  wm = mue(k)/Rp;

  % load vector definition for Hertzian loading
  betam = -3*P*(cos(Rc*wm)*wm*Rc - sin(Rc*wm)) / ...
     (besselj(1,mue(k))^2*Rp^2*Rc^3*wm^3*pi);
  vec = [betam 0 0 0];

  mat(1,1) = (-c*s1+d*s1^3)*exp(-wm*s1*tar.h/2);
  mat(1,2) = (-c*s2+d*s2^3)*exp(-wm*s2*tar.h/2);
  mat(1,3) = (c*s1-d*s1^3)*exp(wm*s1*tar.h/2);
  mat(1,4) = (c*s2-d*s2^3)*exp(wm*s2*tar.h/2);
  mat(2,1) = (-c*s1+d*s1^3)*exp(wm*s1*tar.h/2);
  mat(2,2) = (-c*s2+d*s2^3)*exp(wm*s2*tar.h/2);
  mat(2,3) = (c*s1-d*s1^3)*exp(-wm*s1*tar.h/2);
  mat(2,4) = (c*s2-d*s2^3)*exp(-wm*s2*tar.h/2);
  mat(3,1) = (1-a*s1^2)*exp(-wm*s1*tar.h/2);
  mat(3,2) = (1-a*s2^2)*exp(-wm*s2*tar.h/2);
  mat(3,3) = (1-a*s1^2)*exp(wm*s1*tar.h/2);
  mat(3,4) = (1-a*s2^2)*exp(wm*s2*tar.h/2);
  mat(4,1) = (1-a*s1^2)*exp(wm*s1*tar.h/2);
  mat(4,2) = (1-a*s2^2)*exp(wm*s2*tar.h/2);
  mat(4,3) = (1-a*s1^2)*exp(-wm*s1*tar.h/2);
  mat(4,4) = (1-a*s2^2)*exp(-wm*s2*tar.h/2);

  Co = wm^3*mat\vec';

            %% compute partial derivatives
  % function f(z)
  Fz = Co(1)*exp(s1*wm*z(zi)) + Co(2)*exp(s2*wm*z(zi)) + ...
     Co(3)*exp(-s1*wm*z(zi)) + Co(4)*exp(-s2*wm*z(zi));
  % first der. wrt z of f(z)
  Fp = s1*wm*Co(1)*exp(s1*wm*z(zi)) +...
      s2*wm*Co(2)*exp(s2*wm*z(zi)) -...
      s1*wm*Co(3)*exp(-s1*wm*z(zi)) -...
      s2*wm*Co(4)*exp(-s2*wm*z(zi));
  % second der. wrt z of f(z)
  Fd = s1^2*wm^2*Co(1)*exp(s1*wm*z(zi)) +...
      s2^2*wm^2*Co(2)*exp(s2*wm*z(zi)) +...
      s1^2*wm^2*Co(3)*exp(-s1*wm*z(zi)) +...
      s2^2*wm^2*Co(4)*exp(-s2*wm*z(zi));
  % third der. wrt z of f(z)
  Ft = s1^3*wm^3*Co(1)*exp(s1*wm*z(zi)) +...
      s2^3*wm^3*Co(2)*exp(s2*wm*z(zi)) -...
      s1^3*wm^3*Co(3)*exp(-s1*wm*z(zi)) -...
      s2^3*wm^3*Co(4)*exp(-s2*wm*z(zi));

  % ----------- stress components -----------------------------------%
  SIG_ZZ(zi,rl)=SIG_ZZ(zi,rl)-... .
        c*wm^2*besselj(0,wm*r)*(Fp)+d*besselj(0,wm*r)*(Ft);
```

```
        % due to the fact that one of the terms in the expression for SIG_RR
        % and SIG_TT contains 1/r, those components are not defined at r=0.
        % Therefore a limit process has to be carried out. In the limit for r
        % -> 0 besselj(0,wm*0) = 1 and besselj(1,wm*0) = wm/2.
        if r == 0
          SIG_TT(zi, rl) = SIG_TT(zi,rl)+(b*wm^2*Fp-a*Ft)+(1-b)*wm^2/2*Fp;

          SIG_RR(zi, rl) = SIG_RR(zi,rl)+(wm^2*Fp-a*Ft)+(b-1)*wm^2/2*Fp;
        else
          SIG_TT(zi, rl) = SIG_TT(zi,rl)+besselj(0,wm*r)*...
            (b*wm^2*Fp-a*Ft)+(1-b)*wm*besselj(1,wm*r)/r*Fp;

          SIG_RR(zi, rl) = SIG_RR(zi,rl)+...
            besselj(0,wm*r)*(wm^2*Fp-a*Ft)+(b-1)*wm*besselj(1,wm*r)/r*Fp;
        end

        % check whether we have reached the critical sigma value
        if zi == 1 && SIG_ZZ(1,rl)>SIG_ZZ_b(1,rl)
          kmax = k;
          SIG_ZZ(1,rl) = SIG_ZZ_b(1,rl);
          break
        end

        if zi == 1
          SIG_RZ(zi, rl) = 0;
        else
        SIG_RZ(zi, rl)=SIG_RZ(zi,rl)+...
          wm^3*besselj(1, wm*r)*(Fz)-a*wm*besselj(1, wm*r)*(Fd);
        end

        % update loop variable
        k = k + 1;

      end

    end
    r = r + 0.5*tar.L/resx;
    rl = rl + 1;

  end

  % store the results in the structure. A minus sign has to be included in order to
  % define the stress components properly. Negative values indicate compression.
  SIGC.XX = -SIG_RR(2:end,:);
  SIGC.YY = -SIG_TT(2:end,:);
  SIGC.XZ = -SIG_RZ(2:end,:);
  SIGC.ZZ = -SIG_ZZ(2:end,:);

end

end


%% ------------------------- flex_sigV1.m ------------------------------------ %%
%-------------------------------------------------------------------------------%
% Module for the computation of the flexural stresses during the impact.
%-------------------------------------------------------------------------------%
```

```
% input:
% - current load level, Pt [N]
% - bending compliance matrix, d [Nm^-1]
% - root moment, Mr [Nm] in case of a clamped beam
% - resolution in x direction, resx [-]
% - resolution in z direction, resz [-] (if resz = 1 then we only look at the
% interfaces)
%-------------------------------------------------------------------------------%
% output:
% - structure SIGB containing 5 fields
%     * .xx (2*ni,resx)
%     * .yy (2*ni,resx)
%     * .xy (2*ni, resx)
%     * .xz (ni, resx)
%     * .yz (ni, resx)
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V1
% date: 05.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function[SIGB] = flex_sigV2(Pt, d, Mr, resx, resz)

%% variable declaration
global tar
% applied load on the plate
% I have to come back to this wrt the sign of the applied force
P = Pt;

%% compute flexural stresses in the structure
% compute the z coordiantes of the ply interfaces, starting at the top face
z = flipud(zvector(tar.E0(7), length(tar.seq)));

% x vector for beam model
x = linspace(tar.L/2,0,resx);
%-------------------------------------------------------------------------------%
% in-plane stresses per ply based on laminated beam bending theory (see Reddy)
% this section computes the in-plane stresses due to bending for either simply
% supported or clamped beams (both ends)

% write stiffness matrix for orthotropic plane stress layer
Q_xx = tar.E0(1) / (1 - tar.E0(4)^2 * tar.E0(2)/tar.E0(1));
Q_yy = tar.E0(2) / (1 - tar.E0(4)^2 * tar.E0(2)/tar.E0(1));
Q_xy = tar.E0(4) * tar.E0(2) / (1 - tar.E0(4)^2 * tar.E0(2)/tar.E0(1));
Q_ss = tar.E0(3);

% stress component matrices
SIGB.XX = [];
SIGB.YY = [];
SIGB.XY = [];
SIGB.XZ = [];
SIGB.YZ = [];

% outer loop for the running x coordinate, from boundary to midspan
for xl = 1:length(x)        % xl => x_loop

  % internal bending moment distribution
  M = Mr+P*x(xl)/2;
```

```
dM = P/2;


% initialize the stress values per position x
sig_xx = [];
sig_yy = [];
sig_xy = [];
sig_xz = 0;
sig_yz = 0;

% loop through all plies
for k = 1:1:length(tar.seq)

  % per ply we have a z-vector
  z_ply = z(k)/tar.h:-tar.E0(7)/resz/tar.h:z(k+1)/tar.h;

  % we need the stiffnes matrix Q_bar
  m = cosd(tar.seq(k));
  n = sind(tar.seq(k));

  % transformed plane-stress reduced Stiffness matrix, approach from
  % Kassapoglou's book
  Q(1,1) = (m^4 * Q_xx) + (n^4 * Q_yy) + (2 * m^2 * n^2 * Q_xy)+...
       (4 * m^2 * n^2 * Q_ss);
  Q(2,2) = (n^4 * Q_xx) + (m^4 * Q_yy) + (2 * m^2 * n^2 * Q_xy)+...
       (4 * m^2 * n^2 * Q_ss);
  Q(1,2) = (m^2 * n^2 * Q_xx) + (m^2 * n^2 * Q_yy) + (m^4 + n^4)...
       * Q_xy - (4 * m^2 * n^2 * Q_ss);
  Q(3,3) = (m^2 * n^2 * Q_xx) + (m^2 * n^2 * Q_yy) - ...
       (2 * m^2 * n^2 * Q_xy) + (m^2 - n^2)^2 * Q_ss;
  Q(1,3) = (m^3 * n * Q_xx) - (m * n^3 * Q_yy) + (m*n^3 - m^3*n)...
       * Q_xy + 2 * (m*n^3 - m^3*n) * Q_ss;
  Q(2,3) = (m * n^3 * Q_xx) - (m^3 * n * Q_yy) + (m^3*n - m*n^3)...
       * Q_xy + 2 * (m^3*n - m*n^3) * Q_ss;
  Q(2,1) = Q(1,2);
  Q(3,1) = Q(1,3);
  Q(3,2) = Q(2,3);

  %--------------------------------------------------------------------------%

  % initialize stress values for each ply
  sig_xx_k = [];
  sig_yy_k = [];
  sig_xy_k = [];

  % computation of flexural stresses per ply
  sig_xx_k = z_ply./tar.w*tar.h*M*(Q(1,1)*d(1,1)+Q(1,2)*d(1,2)+Q(1,3)*d(1,3));
  sig_yy_k = z_ply./tar.w*tar.h*M*(Q(1,2)*d(1,1)+Q(2,2)*d(1,2)+Q(2,3)*d(1,3));
  sig_xy_k = z_ply./tar.w*tar.h*M*(Q(1,3)*d(1,1)+Q(2,3)*d(1,2)+Q(3,3)*d(1,3));

%         % transform to ply coordinates
%         SIG_lam_1 = [sig_k(1,:); sig_k(2,:); sig_k(3,:)];
%         % Transformation matrix
%         T = [m^2 n^2 2*m*n; n^2 m^2 -2*m*n; -m*n m*n (m^2-n^2)];
%         SIG_ply_1 = T*SIG_lam_1;

  % store the computed values for the ply in a vector
```

```
    if k == 1
       sig_xx = [sig_xx_k(end)];
       sig_yy = [sig_yy_k(end)];
       sig_xy = [sig_xy_k(end)];
    elseif k == length(tar.seq)
       sig_xx = [sig_xx sig_xx_k(1)];
       sig_yy = [sig_yy sig_yy_k(1)];
       sig_xy = [sig_xy sig_xy_k(1)];
    else
       sig_xx = [sig_xx sig_xx_k];
       sig_yy = [sig_yy sig_yy_k];
       sig_xy = [sig_xy sig_xy_k];
    end

    %-------------------------------------------------------------------------%

    % transverse stresses, integration
    % initialising and updating of the integration constants
    if k == 1
       G = 0;
       F = 0;
    else
       G = sig_xz(end);
       F = sig_yz(end);
    end

    dsig_k_xx = dM/tar.w*(Q(1,1)*d(1,1)+Q(1,2)*d(1,2)+Q(1,3)*d(1,3));
    dsig_k_xy = dM/tar.w*(Q(1,3)*d(1,1)+Q(2,3)*d(1,2)+Q(3,3)*d(1,3));

    sig_xz = [sig_xz -dsig_k_xx*(z_ply(2:end).^2-z_ply(1)^2)*...
          tar.h^2/2 + G];
    sig_yz = [sig_yz -dsig_k_xy*(z_ply(2:end).^2-z_ply(1)^2)*...
          tar.h^2/2 + F];
    sig_zz = 0; % dQ/dx is zero for a concentrated force

  end

  SIGB.XX(:,xl) = -sig_xx';
  SIGB.YY(:,xl) = -sig_yy';
  SIGB.XY(:,xl) = -sig_xy';
  SIGB.XZ(:,xl) = -sig_xz(2:end-1)';
  SIGB.YZ(:,xl) = -sig_yz(2:end-1)';

end

end


function[SIGS] = sing_sigV1(resx, resz, ind)

%% Variable declaration
global tar dam
h = dam.SIF(ind, 4);
eps = dam.SIF(ind, 3);
```

```
KI = dam.SIF(ind, 1);
KII = dam.SIF(ind, 2);
K = (KI+KII*1i)*h^(-1i*eps);
%% define stress field
X = linspace(0, tar.L/2, resx);
Y = linspace(tar.h-tar.E0(7), tar.E0(7), (length(tar.seq)-1)*resz);

% stress components
a = dam.delams(ind, 1);
h1 = dam.delams(ind, 2);
for xl = 1:length(X)
  for yl = 1:length(Y)
    x1 = X(xl) - a;
    y1 = Y(yl) - (tar.h - h1);
    rl = sqrt(x1^2+y1^2);

    % check in which quadrant the coordinate is and adjust theta accordingly
    if x1 > 0 && y1 >= 0
      % 1st quadrant
      tl = atan(y1/x1);
    elseif x1 < 0 && y1 >= 0
      % 2nd quadrant
      tl = atan(y1/x1) + pi;
    elseif x1 < 0 && y1 < 0
      % 3rd quadrant
      tl = atan(y1/x1) - pi;
    elseif x1 > 0 && y1 < 0
      % 4th quadrant
      tl = atan(y1/x1);
    end

    % define parameters and subfunctions for interface crack
    TL = eps*log(rl/(2*a))+tl/2;
    om1 = exp(-eps*(pi-tl));
    om2 = exp(eps*(pi+tl));
    fxx1 = 3*cos(TL)+2*eps*sin(tl)*cos(tl+TL)-sin(tl)*sin(tl+TL);
    fxx2 = 3*sin(TL)+2*eps*sin(tl)*sin(tl+TL)+sin(tl)*cos(tl+TL);
    fzz1 = cos(TL)-2*eps*sin(tl)*cos(tl+TL)+sin(tl)*sin(tl+TL);
    fzz2 = sin(TL)-2*eps*sin(tl)*sin(tl+TL)-sin(tl)*cos(tl+TL);
    fxz1 = sin(TL)+2*eps*sin(tl)*sin(tl+TL)+sin(tl)*cos(tl+TL);
    fxz2 = -cos(TL)-2*eps*sin(tl)*cos(tl+TL)+sin(tl)*sin(tl+TL);

    if y1 > 0
      sigxx(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om1*fxx1-1/om1*cos(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om1*fxx2+1/om1*sin(tl-TL));
      sigzz(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om1*fzz1+1/om1*cos(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om1*fzz2-1/om1*sin(tl-TL));
      sigxz(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om1*fxz1-1/om1*sin(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om1*fxz2-1/om1*cos(tl-TL));
    elseif y1 < 0
      sigxx(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om2*fxx1-1/om2*cos(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om2*fxx2+1/om2*sin(tl-TL));
      sigzz(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om2*fzz1-1/om2*cos(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om2*fzz2+1/om2*sin(tl-TL));
      sigxz(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om2*fxz1-1/om2*cos(tl-TL))...
        -KII/(2*sqrt(2*pi*rl))*(om2*fxz2+1/om2*sin(tl-TL));
```

```
    else
      if x1 > 0
        tl = 0;
        sigzz(yl, xl) = real(K*rl^(1i*eps)/sqrt(2*pi*rl));
        sigxz(yl, xl) = imag(K*rl^(1i*eps)/sqrt(2*pi*rl));
        sigxx(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om1*fxx1-1/om1*cos(tl-TL))...
          -KII/(2*sqrt(2*pi*rl))*(om1*fxx2+1/om1*sin(tl-TL));
      else
        tl = pi;
        sigzz(yl, xl) = 0;
        sigxz(yl, xl) = 0;
        sigxx(yl,xl) = KI/(2*sqrt(2*pi*rl))*(om1*fxx1-1/om1*cos(tl-TL))...
          -KII/(2*sqrt(2*pi*rl))*(om1*fxx2+1/om1*sin(tl-TL));
      end

    end

  end

end

SIGS.XX = sigxx;
SIGS.ZZ = sigzz;
SIGS.XZ = sigxz;


%% --------------------------- fail_analysisV2.m ---------------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function simply applies the selected failure criterion (eq. 3.6.3) to the
% local stress valuesand determines whether or not a "cell" has failed. However, a
% delamination will only be recognized if it fulfills certain requirements: it must
% not lie between equally oriented plies and the failed section must be longer than
% 4-5 ply thicknesses, or equivalently 2-2.5 times for half a delamination.
%-------------------------------------------------------------------------------------%
% input:
% - structure SIGPLY containing the following matrices with stress components in ply
% coordinate system:
%     * S1, S3, S4, S5
% - current indentation value, alphat
%-------------------------------------------------------------------------------------%
% output:
% - failflag, indicates whether a delamination can be recognized
% - failinfo, gives position of the delamination within the stack as well as length
%   from the centre
% - FAIL, failure indication matrix, provides a visual overview of the failed cells.
% Writes 1 if the cell has failed otherwise remains 0
%-------------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V2
% date: 08.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function[flag, failinfo, FAIL] = fail_analysisV2(SIGPLY, alphat)

%% variabel declaration
global tar sim resx
```

```
%% Check ply interfaces for delaminations
% strength data taken from Hexcel for IM7/8551-7. Only exception is Z_C, which has
% been taken from Lopes' thesis (table 6.1) assuming that X_(2-) = Z_C.
Z_C = 200e6;              % transverse compressive strength,[Pa]
S = 100e6;               % short-beam shear strength, [Pa]
X_T = 2757e6;             % tensile in-plane strength, [Pa]
X_C = 1620e6;             % compressive in-plane strength, [Pa]

% r-vector
r = linspace(0,tar.L/2,resx);

% creating failure matrix
FAIL = zeros(length(tar.seq)-1, size(SIGPLY.S1,2));
failinfo = [];
countk = 1;
for k = 1:length(tar.seq)-1
  for rl = 1:size(SIGPLY.S1,2)

    % according to Brewer-Lagace, modified to take into account the in-plane
    % stress
    % We have to check each interface twice. If one of them indicates a failure
    % that is already considered a failed interface then.
    % upper
    if SIGPLY.S1(2*(k-1)+1,rl) < 0
      checku =(SIGPLY.S3(2*(k-1)+1,rl)/Z_C)^2+(SIGPLY.S4(2*(k-1)+1,rl)/S)^2 ...
          +(SIGPLY.S5(2*(k-1)+1,rl)/S)^2+(SIGPLY.S1(2*(k-1)+1,rl)/X_C)^2;
    else
      checku =(SIGPLY.S3(2*(k-1)+1,rl)/Z_C)^2+(SIGPLY.S4(2*(k-1)+1,rl)/S)^2 ...
          +(SIGPLY.S5(2*(k-1)+1,rl)/S)^2+(SIGPLY.S1(2*(k-1)+1,rl)/X_T)^2;
    end
    % lower
    if SIGPLY.S1(2*(k-1)+2,rl) < 0
      checkl =(SIGPLY.S3(2*(k-1)+2,rl)/Z_C)^2+(SIGPLY.S4(2*(k-1)+2,rl)/S)^2 ...
          +(SIGPLY.S5(2*(k-1)+2,rl)/S)^2+(SIGPLY.S1(2*(k-1)+2,rl)/X_C)^2;
    else
      checkl =(SIGPLY.S3(2*(k-1)+2,rl)/Z_C)^2+(SIGPLY.S4(2*(k-1)+2,rl)/S)^2 ...
          +(SIGPLY.S5(2*(k-1)+2,rl)/S)^2+(SIGPLY.S1(2*(k-1)+2,rl)/X_T)^2;
    end

    if checku > 1 || checkl > 1
      % demand that delamination lies between unequally oriented plies and is
      % longer than 2.5tply. Additionally, if k = 1 then the indentation must
      % not be larger than a single tply
      if tar.seq(k) ~= tar.seq(k+1) && r(rl) > 2.5*tar.E0(7)
        if (k == 1 && alphat < tar.E0(7)) || k ~= 1
          FAIL(k,rl) = 1;    % 1 -> interface fails
          if isempty(failinfo)
            failinfo(countk,:) = [r(rl) k*tar.E0(7) 0];
          elseif k*tar.E0(7) == failinfo(countk,2)
            failinfo(countk,:) = [r(rl) k*tar.E0(7) 0];
          else
            countk = countk + 1;
            failinfo(countk,:) = [r(rl) k*tar.E0(7) 0];
          end
          % we are only interested in the "earliest" (closest to the impact
          % site) occurrence of failure
```

```
            if sim.br == 1
               break
            end
          end
        end
      end

      [I,J] = find(FAIL);
      if isempty(I)
        flag = 0;
      else
        flag = 1;
      end
    end
end

end


%% ------------------------- LEFM_toolV3.m ------------------------------------ %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the linear elastic fracture mechanics tool, the main module of the
% simulation. It covers the following aspects:
% - Setting up the interface matrix (with "interface_func")
% - Computing the segment and section properties
% - Determining the displacement functions of all the segments/sections (with
% "disp_func")
% - applying a filtering scheme to determine which segments belong to the computation
% of a certain delamination
% - computing the Energy Release Rate using the strain energy integral (with
% "ERR_strain")
% - determining whether any of the delaminations is growing?
% - Computing the Stress Intensity Factor
% - Updating the bending compliance
%-------------------------------------------------------------------------------%
% inputs:
% - current load, Pt
% - con, contact flag: 0 -> unconstrained; 1-> constrained
% - by declaring the variable "dam" globally it enables access to the structure
% defined in the master file containing the field "delams"
% - delams: matrix with size and thickness positions of all delaminations. The
% through-the-thickness position is given in meters rather than interface number
% outputs:
% - several fields will be added to the structure "dam"
%     * deq - equivalent bending compliance
%     * SIF - stress intensity factors, per tip 2 values, KI and KII
%     * ERR - matrix with ERR values, per tip 2 values, unconstrained and constrained
%             approach
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V1
% date: 08.11.12
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Start of function
function[Pnew] = LEFM_toolV3(P,wc,contact)

%% Variable declaration
global tar dam
```

```
% define the output variable. Only in case of growth will it change.
Pnew = P;

%% input
% call interface function in order to create the interface matrix and obtain all
% sublaminates
[interface, tips, seqk, height, elemcount] = interface_funcV3(dam.delams);
dam.delams = tips;
seqk{elemcount} = tar.seq;
height(elemcount) = tar.h;

%% segment and beam properties
% for each beam segement the reduced bending stiffness matrix needs to be computed.
for el = 1:elemcount
  [K, ~] = ABDmat(tar.E0, seqk{el}, 0, 0);
  % reduced bending matrix D due to unsymmetric and unbalanced laminates
  Dred = K(4:6,4:6);
  % compliance matrix
  dred = inv(Dred);
  % Youngs modulus of the composite plate, based on bending stiffness (see
  % Kassapoglou or Reddy)
%         E(el) = 12/(height(el)^3*dred(1,1));
  EI(el) = tar.w/dred(1,1);
end

% section flexural rigidity for constrained contact
n = size(interface,2) + 1;
for x = 1:n-1
 EIsec(x) = sum(EI(unique(interface(:,x))));
end
EIsec(n) = EI(end);

% x-coordinate for the crack tips
xc = tar.L/2-flipud(unique(tips(:,1)));

%% beam displacement
% solve the system of equations resulting from the unknown segment and section
% displacements, based on the chosen contact formulation
if contact == 0
  Cmat = SoE_ucbV2(interface,EI,xc,P);
else
  Cmat = SoE_cbV2(interface,EIsec,xc,P);
end

%% ERR and SIF
for gl = 1:size(tips,1)
  % ERR and SIF, calling ERR_Strain
  [dam.G(gl,1),dam.SIF(gl,1:5)]=ERR_strainV5(tips,gl,contact,P);
  if dam.G(gl,1) > tar.Gcr
    dam.delams(gl,3) = 1;
    % calling delam_growth in order to determine new delamination size, P, G and
    % SIF.
%         [Pnew,dam.G(gl,1),dam.SIF(gl,1:5)] = delam_growth_appdispV2(gl,contact,wc);
    % alternatively, only update delamination size and P
    [Pnew, ~, ~] = delam_growth_appdispV2(gl,contact,wc);
    fprintf(1, 'the new applied load is %1.4e N.\n', Pnew)
  end
end
```

```
%% update equivalent bending compliance

% write representative d(1,1) entry for simply supported or clamped beam
% (see p. 97, notebook)
if tar.BC == 0
   ds11 = 48*tar.w*wc/(Pnew*tar.L^3);
else
   ds11 = 192*tar.w*wc/(Pnew*tar.L^3);
end
[K,~] = ABDmat(tar.E0, tar.seq, 0, 0);
D = K(4:6,1:3);
d = inv(D);
di = 1-ds11/d(1,1);
dam.deq = (1-di)*d;

end


%% -------------------------- Pupdate_func.m ------------------------------------ %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function to determine new applied load, once new delamination sites have been
% found.
%-------------------------------------------------------------------------------%
% input:
% - wc, current applied displacment
%
% output:
% - Pnew, updated applied force due to the change in compliance
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% date: 15.11.2012
% Version: V1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of the function
function[Pnew] = Pupdate_funcV1(wc, contact)

%% Variable declaration
% the global variable "dam" and "tar" will be used to pass information between the
% LEFM_tool and the master file
global dam tar

%% input
% call interface function in order to create the interface matrix and obtain all
% sublaminates
[interface, tips, seqk, height, elemcount] = interface_funcV3(dam.delams);
seqk{elemcount} = tar.seq;
height(elemcount) = tar.h;

%% segment and beam properties
% for each beam segement the reduced bending stiffness matrix needs to be computed.
for el = 1:elemcount
   [K, ~] = ABDmat(tar.E0, seqk{el}, 0, 0);
   % reduced bending matrix D due to unsymmetric and unbalanced laminates
   Dred = K(4:6,4:6);
   % compliance matrix
   dred = inv(Dred);
```

```
  % Youngs modulus of the composite plate, based on bending stiffness (see
  % Kassapoglou or Reddy)
%         E(el) = 12/(height(el)^3*dred(1,1));
  EI(el) = tar.w/dred(1,1);
end

% section flexural rigidity for constrained contact
n = size(interface,2) + 1;
for x = 1:n-1
  EIsec(x) = sum(EI(unique(interface(:,x))));
end
EIsec(n) = EI(end);

% x-coordinate for the crack tips
xc = tar.L/2-flipud(unique(tips(:,1)));

%% beam displacement
% solve the system of equations resulting from the unknown segment and section
% displacements, based on the chosen contact formulation.
% This will serve as the new compliance (see page 31 in NB 3). Use unit load as input
if contact == 0
  Cmat = SoE_ucbV2(interface,EI,xc,1);
else
  Cmat = SoE_cbV2(interface,EIsec,xc,1);
end

% displacement due to unit load
wc_u = Cmat(1,end-1) + Cmat(2,end-1)*tar.L/2 +...
    Cmat(3,end-1)*(tar.L/2)^2 + Cmat(4,end-1)*(tar.L/2)^3;

% new applied load
Pnew = wc/wc_u;

end


%% -------------------------- interface_func.m ---------------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function determines the interface based on the numbering scheme adopted in
% this work.
%-------------------------------------------------------------------------------%
% input:
% - matrix delams, containing through-the-thickness (z) position and length of the
% delamination. The z-coordinate must be given in meters
%-------------------------------------------------------------------------------%
% output:
% - interface, matrix containing the numbering and connectivity of the separated
% beam segments. The number of columns represents the number of delaminations of
% different length
% - tips, same as delams, only now it is sorted according to:
%     * 1st the length (from long to short) and
%     * 2nd the position within the stack (from impact location to back face)
% - seqk, the sublaminates created due to delaminations
% height, vector containing the height of the beam segments
% elemcount, number of beam segments including the undamaged part. "elemcount" also
% serves as indicator to this part
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V1
% date: 08.11.2012
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function [interface, tips, seqk, height, elemcount] = interface_funcV3(delams)

%% Variable declaration
global tar
h_pl = tar.h;
seq =tar.seq;
E0 = tar.E0;

%% Preliminaries
% sort the delaminations according to their length for the numbering of the segments
if size(delams,1) > 1
  % sorted matrix of delamination size and location
  [~,i] = sort(delams, 1, 'descend');
  for ind = 1:size(i,1);
    tips(ind,:) = delams(i(ind),:);
  end
else
  tips = delams;
end

[D, ~, Ic] = unique(tips(:,1));
for ind = 1:length(D)
  I = find(tips==D(ind));
  if length(I) > 1
    tips(I(1):I(end),:) = sort(tips(I(1):I(end),:),1);
  end
end

% get the integer interface numbers
tipseq(:,1) = round(tips(:,2)/E0(7));

% number of delaminations
n = size(tips,1);
% initiation of the interface matrix. It serves as input for the continuity equations
% to be set up for solving for the unknowns.
interface = [1;2];

elemcount = 1;
% element pointer for interface creation
count = 3;
% flags for interface creation
att1 = 0;
att2 = 0;

%% numbering scheme
for ind = 1:n
  % determining where in the beam segment the next smaller delamination splits the
  % segment into two new ones
  if n == 1
    height(elemcount:elemcount+1) = [tips(1,2) h_pl-tips(1,2)];
    seqk{elemcount} = seq(1:tipseq(1,1));
    seqk{elemcount+1} = seq(tipseq(1,1)+1:end);
```

```
    elemcount = elemcount + 2;
    newseg = 1;

  else
    % check how many equal sized delaminations exist
    I = find(tips(:,1) == tips(ind,1));
    if length(I) > 1
      % is current delamination size (ind) the first of its length (compare to
      % I)
      if ind == I(1)
        if ind == 1
          height(elemcount) = tips(1,2);
          seqk{elemcount} = seq(1:tipseq(1,1));
          elemcount = elemcount + 1;
          newseg = 1;
        else
          for x = 1:ind
            tipz = tips(ind,2);
            tipz2 = round(tipz/E0(7));
            tipres = sort(tips(1:ind-1,2));
            tipres2 = round(tipres/E0(7));
            if x == 1
              if (0 < tipz) && (tipz < tipres(1))
                newseg = 1;
                height(elemcount) = tipz;
                seqk{elemcount} = seq(1:tipz2);
                elemcount = elemcount + 1;
                if tips(ind+1,2) > tipres(1)
                  height(elemcount) = tipres(1)-tipz;
                  seqk{elemcount} = seq(tipz2+1:tipres2(1));
                  elemcount = elemcount + 1;
                  att1 = 1;
                end
                break
              end
            elseif x == ind
              if (tipres(end) < tipz) && (tipz < tips(ind+1,2))
                newseg = ind;
                height(elemcount) = tipz-tipres(end);
                seqk{elemcount} = seq(tipres2(end)+1:tipz2);
                elemcount = elemcount + 1;
              end
            elseif (tipres(x-1) < tipz) && (tipz < tipres(x))
              newseg = x;
              height(elemcount) = tipz-tipres(x-1);
              seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
              elemcount = elemcount + 1;
              att2 = 1;
              if tips(ind+1,2) > tipres(x)
                height(elemcount) = tipres(x) - tipz;
                seqk{elemcount} = seq(tipz2+1:tipres2(x));
                elemcount = elemcount + 1;
                att1 = 1;
                att2 = 0;
              end
```

```
            break
          end
        end
      end
    elseif ind == I(end)
      for x = 2:ind
        tipz = tips(ind,2);
        tipz2 = round(tipz/E0(7));
        [tipres, xind] = sort(tips(1:ind-1,2));
        tipres2 = round(tipres/E0(7));
        if x == ind
          if (tipres(end) < tipz) && (tipz < h_pl)
            newseg = ind;
            height(elemcount:elemcount+1) = [tipz-tipres(end)...
                                    h_pl-tipz];
            seqk{elemcount} = seq(tipres2(end)+1:tipz2);
            seqk{elemcount+1} = seq(tipz2+1:end);
            elemcount = elemcount + 2;
            temp = unique(tips(xind,1));
            if length(temp) > 1
              att1 = 1;
            else
              att1 = 0;
            end
          elseif (tipres(end) > tipz)
            newseg = ind-1;
            height(elemcount:elemcount+1) = [tipz-tipres(end-1)...
                                    tipres(end)-tipz];
            seqk{elemcount} = seq(tipres2(end-1)+1:tipz2);
            seqk{elemcount+1} = seq(tipz2+1:tipres2(end));
            elemcount = elemcount + 2;
            att1 = 1;
          end
        elseif (tipres(x-1) < tipz) && (tipz < tipres(x))
          newseg = x;
          height(elemcount:elemcount+1) = [tipz-tipres(x-1)...
                                  tipres(x)-tipz];
          seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
          seqk{elemcount+1} = seq(tipz2+1:tipres2(x));
          elemcount = elemcount + 2;
          att3 = 1;
          break
        end
      end
    else
      quit = 0;
      tipz = tips(ind,2);
      tipz2 = round(tipz/E0(7));
      [tipres, xind] = sort(tips(1:ind-1,2));
      tipres2 = round(tipres/E0(7));
      for x = 2:ind
        if (tips(ind,1) == tips(xind(x-1),1))
          if x == ind || (tips(ind+1,2) < tipres(x))
```

```
            quit = 0;
            break
        end


%                       keyboard
    elseif x == ind
        % the current delamination is either situated beneath or
        % above a longer one or between two of the same length as the
        % current one
        if (tipres(x-1)<tipz) && (tipres(x-1)>tips(ind-1,2))...
            && (tips(ind,1) ~= tips(xind(x-1),1))
            att1 = 1;
            newseg = ind;
            height(elemcount) = tipz - tipres(x-1);
            seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
            elemcount = elemcount + 1;
            quit = 1;
            break
        else
            newseg = x-1;
            height(elemcount:elemcount+1) = [tipz - tips(ind-1,2) ...
                                tipres(x-1) - tipz];
            seqk{elemcount} = seq(tipseq(ind-1,1)+1:tipz2);
            seqk{elemcount+1} = seq(tipz2+1:tipres2(x-1));
            elemcount = elemcount + 2;
            quit = 1;
            att3 = 1;
            break
        end

    % current delam lies in-between two different sized delaminations
    elseif (tipres(x-1) < tipz) && (tipres(x-1) > tips(ind-1,2)) ...
            && (tips(ind,1) ~= tips(xind(x-1),1)) && ...
            (tipres(x) > tipz) && (tipres(x) < tips(ind+1,2))...
            && (tips(ind,1) ~= tips(xind(x),1))
        newseg = x;
        height(elemcount:elemcount+1) = [tipz - tipres(x-1) ...
                            tipres(x)-tipz];
        seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
        seqk{elemcount+1} = seq(tipz2+1:tipres2(x));
        elemcount = elemcount + 2;
        quit = 1;
        att3 = 1;
        break

    % current delam lies below a longer exisiting one
    elseif (tipres(x-1) < tipz) && (tipres(x-1) > tips(ind-1,2))...
            && (tips(ind,1) ~= tips(xind(x-1),1))...
            && (tipres(x) > tipz)
        newseg = x;
        height(elemcount) = tipz - tipres(x-1);
        seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
        elemcount = elemcount + 1;
        quit = 1;
        att1 = 1;
```

```
            break
          % current delam lies above a longer exisiting one
          elseif (tipres(x-1) > tipz) && (tipres(x-1) < tips(ind+1,2))...
              && (tips(ind,1) ~= tips(xind(x-1),1))
            newseg = x-1;
            height(elemcount:elemcount+1) = [tipz - tips(ind-1,2) ...
                                tipres(x-1) - tipz];
            seqk{elemcount} = seq(tipseq(ind-1,1)+1:tipz2);
            seqk{elemcount+1} = seq(tipz2+1:tipres2(x-1));
            elemcount = elemcount + 2;
            quit = 1;
            att3 = 1;
            break
          end
        end
        % current delamination lies between two equal length
        if quit == 0
          newseg = newseg + 1;
          height(elemcount) = tipz-tips(ind-1,2);
          seqk{elemcount} = seq(tipseq(ind-1,1)+1:tipz2);
          elemcount = elemcount + 1;
          temp = unique(tips(xind,1));
          if length(temp) > 1
            att1 = 1;
          else
            att1 = 0;
          end
        end
      end
    else
      % not equal sized than the first one
      for x = 1:ind
        if ind == 1
          height(elemcount:elemcount+1) = [tips(1,2) h_pl-tips(1,2)];
          seqk{elemcount} = seq(1:tipseq(1,1));
          seqk{elemcount+1} = seq(tipseq(1,1)+1:end);
          elemcount = elemcount + 2;
        else
          tipz = tips(ind,2);
          tipz2 = round(tipz/E0(7));
          tipres = sort(tips(1:ind-1,2));
          tipres2 = round(tipres/E0(7));
          if x == 1
            if (0 < tipz) && (tipz < tipres(1))
              newseg = 1;
              height(elemcount:elemcount+1) = [tipz tipres(1)-tipz];
              seqk{elemcount} = seq(1:tipz2);
              seqk{elemcount+1} = seq(tipz2+1:tipres2(1));
              elemcount = elemcount + 2;
            end
          elseif x == ind
            if (tipres(end) < tipz) && (tipz < h_pl)
              newseg = ind;
```

```
                height(elemcount:elemcount+1) = [tipz-tipres(end)...
                                    h_pl-tipz];
                seqk{elemcount} = seq(tipres2(end)+1:tipz2);
                seqk{elemcount+1} = seq(tipz2+1:end);
                elemcount = elemcount + 2;
              end
            else
              if (tipres(x-1) < tipz) && (tipz < tipres(x))
                newseg = x;
                height(elemcount:elemcount+1) = [tipz-tipres(x-1)...
                                    tipres(x)-tipz];
                seqk{elemcount} = seq(tipres2(x-1)+1:tipz2);
                seqk{elemcount+1} = seq(tipz2+1:tipres2(x));
                elemcount = elemcount + 2;
              end
            end
          end
        end
      end
    end
%       keyboard
     %% updating the interface matrix
    if ind > 1
      if newseg == ind
        if tips(ind,1) == tips(ind-1,1)
          if (att1 == 1 && ind == I(end))
            interface = [interface(1:newseg-1,:);
                    [interface(newseg,1:end-1) count];
                    interface(newseg, 1:end-1) count+1];
            count = count + 2;
            att1 = 0;
          elseif att1 == 1
            interface = [interface(1:newseg-1,:);
                    [interface(newseg,1:end-1) count];
                    interface(newseg:end,:)];
            count = count + 1;
            att1 = 0;
          else
            interface = [interface(1:newseg,:);
                    interface(end,1:end-1) count];
            count = count + 1;
          end
        elseif length(I) == 1
          interface = [interface(1:newseg,:) [interface(1:newseg-1, end);...
                    count];
                    interface(newseg,:) count+1];
          count = count + 2;
        else
          interface = [interface(1:newseg,:) [interface(1:newseg-1, end);...
                   count];
                    interface(newseg,:) count];
          count = count + 1;
        end

      else
```

```
        if tips(ind,1) == tips(ind-1,1)
          if att1 == 1 && ind == I(end)
            interface = [interface(1:newseg-1,:);
                    [interface(newseg,1:end-1) count];
                    [interface(newseg, 1:end-1) count+1];
                    interface(newseg+1:end,:)];
            count = count + 2;
            att1 = 0;
          elseif att1 == 1
            interface = [interface(1:newseg-1,:);
                    [interface(newseg,1:end-1) count];
                    interface(newseg:end,:)];
            count = count + 1;
            att1 = 0;

          elseif att3 == 1
            interface = [interface(1:newseg-1,:);
                    [interface(newseg,1:end-1) count];
                    [interface(newseg, 1:end-1) count+1];
                    interface(newseg+1:end,:)];
            count = count + 2;
            att3 = 0;
          end
        else
          if att1 == 1 || length(I) == 1
%                         fprintf(1, 'hah\n')
            interface = [interface(1:newseg,:) [interface(1:newseg-1,end);count];
                    [interface(newseg,:) count+1];
                    interface(newseg+1:end,:) interface(newseg+1:end,end)];
            count = count + 2;
            att1 = 0;
          else
%                         fprintf(1,'heh\n')
            interface = [interface(1:newseg,:) [interface(1:newseg-1,end);count];
                    [interface(newseg,:) count];
                    interface(newseg+1:end,:) interface(newseg+1:end,end)];
            count = count + 1;
          end
        end
      end
    end
end

end
```

```
function Cmat = SoE_cbV2(interface, EIsec, xc, P)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function solves for the displacement function of the beam segment assuming
% constrained contact, i.e. parallel beam segments have the same displacement.
% ------------------------------------------------------------------------------ %
% Input:
% - interface (matrix showing the segment connectivity)
% - EIsec (vector containing flexural rigidities for each section defined by a change
% in the "cross-section" of the beam)
% - xc (vector containg the x-coordinate of the delamination tips)
% - BC (boundary condition: 0 -> simply supported; 1 -> clamped)
% - P (current load acting on the beam)
% ------------------------------------------------------------------------------ %
% Output:
% - Cmat (matrix of beam displacement coefficients where every column represents
% another segment and the displacment per segment is defined as:
% w_i(x) = Cmat(1,i) + Cmat(2,i)*x + Cmat(3,i)*x^2 + Cmat(4,i)*x^3;

% file info:
% - author: J.J. Kurpierz
% - date: 16.10.2012
% - version: V1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% variable declaration
global tar
L = tar.L;
BC = tar.BC;

%% System of 4*(elemcount)-2 equations
% initialize matrix, force vector and row count
% number of segments
n = size(interface,2)+1;
elemcount = length(unique(interface))+1;
% the number of unknowns per section is two due to the consecutive integration of the
% expression d2w/dx2 = -M/EI. The total number of unknowns is therfore 2*n. In case
% of the clamped beam we have the unknown root moment to account for resulting in
% 2n+1 unknowns.
% However, in the case of a clamped beam, the first section (the undamaged one) will
% always have the the two integration constants equal to zero. Therefore we have 2n-1
% equations in the system of equations. For the simply supported there is one
% coefficient for the undamaged part that can be dropped, again resulting in 2n-1
% equations, i.e. a solvable system.
A = zeros(2*n-1, 2*n-1);
F = zeros(2*n-1,1);
% intialize the row count variable
y = 1;

%---------------------------- boundary conditions ----------------------------------%
% boundary formulations for the clamped and simply supported beam
% The conditions at the support of the beam (x=0) do not need to be taken into
% acount since it would yield in one or two rows where all entries are equal to zero

% at midspan of the beam, x = L/2
% rotation of the beam is equal to 0, w_n-1' = 0
switch BC
  case 0
    A(y,2*(n-1)-1) = 1;
    F(y) = P*L^2/(16*EIsec(n-1));
  case 1
    A(y,2*(n-1)-1) = 1;
    A(y, end) = -L/(2*EIsec(n-1));
```

```
    F(y) = P*L^2/(16*EIsec(n-1));
end



%--------------------------- continuity equations -------------------------------%

% at each interface boundary (at each entry of xc) we have two continuity equations:
% displacement and rotation are equal to each other. Only between the undamaged and
% first damaged part we have to apply slightly different conditions


% go through each crack
for x = 1:n-1
  switch BC
    case 0
      if x == 1
        % displacement
        % update count variable
        y = y+1;
        A(y, 1:2) = -[xc(1) 1];
        A(y, end) = xc(1);
        F(y) = P*xc(1)^3/12*(1/(EIsec(n)) - 1/(EIsec(1)));

        % rotation
        % update count variable
        y = y+1;
        A(y,1) = -1;
        A(y, end) = 1;
        F(y) = P*xc(1)^2/4*(1/(EIsec(n)) - 1/(EIsec(1)));

      else
        % displacement
        % update count variable
        y = y+1;
        A(y, 2*(x-1)-1:2*x) = [xc(x) 1 -xc(x) -1];
        F(y) = P*xc(x)^3/12*(1/(EIsec(x-1)) - 1/(EIsec(x)));

        % rotation
        % update count variable
        y = y+1;
        A(y, 2*(x-1)-1:2*x) = [1 0 -1 0];
        F(y) = P*xc(x)^2/4*(1/(EIsec(x-1)) - 1/(EIsec(x)));
      end
    case 1
      if x == 1
        % displacement
        % update count variable
        y = y+1;
        A(y, 1:2) = -[xc(1) 1];
        A(y, end) = xc(1)^2/2*(1/(EIsec(1)) - 1/(EIsec(n)));
        F(y) = P*xc(1)^3/12*(1/(EIsec(n)) - 1/(EIsec(1)));

        % rotation
        % update count variable
        y = y+1;
```

```
            A(y,1) = -1;
            A(y, end) = xc(1)*(1/(EIsec(1)) - 1/(EIsec(n)));
            F(y) = P*xc(1)^2/4*(1/(EIsec(n)) - 1/(EIsec(1)));

        else
            % displacement
            % update count variable
            y = y+1;
            A(y, 2*(x-1)-1:2*(x-1)) = [xc(x) 1];
            A(y, 2*x-1:2*x) = -[xc(x) 1];
            A(y, end) = xc(x)^2/2*(1/(EIsec(x)) - 1/(EIsec(x-1)));
            F(y) = P*xc(x)^3/12*(1/(EIsec(x-1)) - 1/(EIsec(x)));

            % rotation
            % update count variable
            y = y+1;
            A(y, 2*(x-1)-1) = 1;
            A(y, 2*x-1) = -1;
            A(y, end) = xc(x)*(1/(EIsec(x)) - 1/(EIsec(x-1)));
            F(y) = P*xc(x)^2/4*(1/(EIsec(x-1)) - 1/(EIsec(x)));
        end

    case 2
        if x == 1
            % displacement
            % update count variable
            y = y+1;
            A(y, 1:2) = -[xc(1) 1];
            F(y) = P*(L/2*xc(1)^2-xc(1)^3/6)*(1/(EIsec(1)) - 1/(EIsec(n)));

            % rotation
            % update count variable
            y = y+1;
            A(y,1) = -1;
            F(y) = P*(L*xc(1) - xc(1)^2/2)*(1/(EIsec(1)) - 1/(EIsec(n)));

        else
            % displacement
            % update count variable
            y = y+1;
            A(y, 2*(x-1)-1:2*x) = [xc(x) 1 -xc(x) -1];
            F(y) = P*(L/2*xc(1)^2-xc(1)^3/6)*(1/(EIsec(x)) - 1/(EIsec(x-1)));

            % rotation
            % update count variable
            y = y+1;
            A(y, 2*(x-1)-1:2*x) = [1 0 -1 0];
            F(y) = P*(L*xc(1) - xc(1)^2/2)*(1/(EIsec(x)) - 1/(EIsec(x-1)));
        end

    end
end

if BC == 2
    % remove first row and last column
    A(1,:) = [];
    A(:,end) = [];
```

```
    F(1) = [];
end

%% solve the system
C1 = A\F;
% SVD approach
tol = size(A,2)*2^-53;
[U,S,V] = svd(A);
Sinv = zeros(size(A,2));
for il = 1:size(A,2)
    if (S(il,il)/S(1) < tol) | (S(il,il) == 0)
        Sinv(il,il) = 0;
    else
        Sinv(il,il) = 1/S(il,il);
    end
end

if size(A,1) == size(A,2)
    % solution vector according to SVD
    C = V*Sinv*U'*F;
else
    C = 0;
    for il = 1:size(A,2)
        C = C + dot(U(:,il),F)*Sinv(il,il)*V(:,il);
    end
end

%% rearranging the coefficients, making them easier to discern
Cmat = zeros(4,n);
for ind = 1:n
    if ind == n
        switch BC
            case 0
                Cmat(:,ind) = [0; C(end); 0; -P/(12*EIsec(n))];
            case 1
                Cmat(:,ind) = [0;0; -C(end)/(2*EIsec(n)); -P/(12*EIsec(n))];
            case 2
                Cmat(:,ind) = [0;0; P*L/(2*EIsec(n)); -P/(6*EIsec(n))];
        end
    else
        switch BC
            case 0
                Cmat(1:4,ind) = [C(2*ind); C(2*ind-1); 0; -P/(12*EIsec(ind))];
            case 1
                Cmat(1:4,ind) = [C(2*ind); C(2*ind-1); ...
                    -C(end)/(2*EIsec(ind)); -P/(12*EIsec(ind))];
            case 2
                Cmat(1:4,ind) = [C(2*ind); C(2*ind-1); ...
                    P*L/(2*EIsec(ind)); -P/(6*EIsec(ind))];
        end
    end
end
```

```
function [Cmat] = SoE_ucbV2(interface, EI, xc, P)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function solves for the displacement function of the beam segments assuming
% unconstrained contact, i.e. each beam segment is moving indenpendently.
% ------------------------------------------------------------------------------ %
% Input:
% - interface (matrix showing the segment connectivity)
% - EI (vector containing flexural rigidities for each beam segment)
% - xc (vector containg the x-coordinate of the delamination tips)
% - BC (boundary condition: 0 -> simply supported; 1 -> clamped)
% - P (current load acting on the beam)
% ------------------------------------------------------------------------------ %
% Output:
% - Cmat (matrix of beam displacement coefficients where every column represents
% another segment and the displacment per segment is defined as:
% w_i(x) = Cmat(1,i) + Cmat(2,i)*x + Cmat(3,i)*x^2 + Cmat(4,i)*x^3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% file info
% author: JJ Kurpierz
% date: 16.10.2012
% version: V1

%% variable declaration
global tar
L = tar.L;
BC = tar.BC;

%% System of 4*(elemcount)-2 equations
% initialize matrix, force vector and row count
elemcount = length(EI);
elems = 1:elemcount;
A = zeros(4*elemcount-2, 4*elemcount-2);
F = zeros(4*elemcount-2,1);
% intialize the row count variable
y = 1;
% number of delaminations
n = size(interface,1) - 1;

%% ------------------------- boundary conditions ----------------------------- %%

switch BC

  case 0
    % boundary formulations for the simply supported beam
    % For the case of a simply supported beam, we have the following two
    % conditions at the support: - deflection w is equal to zero -> C_0,1 = 0 and
    % the moment is equal to zero -> C_0,3 = 0. These two coefficients do not
    % have to be taken into account anymore. This means that in the case of a
    % simply supported beam the last two entries of C are C_0,2 and C_0,4.

    % at midspan of the beam, x = L/2
    % applied force P/2 equals the sum of the shear forces V_k acting in the
    % segments. V_k = M_k' = -EI_k*w_k''' = -EI_k(6*C_k,4)
    for x = 1:n+1
      A(y,interface(x,end)*4) = -6*EI(interface(x,end));
    end
    F(1) = P/2;

    % the rotation of the beam segments is equal to zero, w_k' = 0
```

```
    for x = 1:n+1
        y = y+1;
        A(y,4*(interface(x,end)-1)+2:4*interface(x,end)) =...
            [1, L, 3*(L/2)^2];
    end

    % the displacements of the beam segements are the same at the end of the
    % clamped, w_k = w_k+1
    for x = 1:n
        y = y + 1;
        A(y,4*(interface(x,end)-1)+1:4*interface(x,end)) = ...
            [1, L/2, (L/2)^2, (L/2)^3];
        A(y,4*(interface(x+1,end)-1)+1:4*interface(x+1,end)) =...
            -[1, L/2, (L/2)^2, (L/2)^3];
    end

case 1
    % boundary formulations for the clamped beam
    % The conditions at the root of the beam (x=0) does not need to be taken into
    % acount since it would yield in two rows where all entries are equal to zero

    % at midspan of the beam, x = L/2
    % applied force P/2 equals the sum of the shear forces V_k acting in the
    % segments. V_k = M_k' = -EI_k*w_k''' = -EI_k(6*C_k,4)
    for x = 1:n+1
        A(y,interface(x,end)*4) = -6*EI(interface(x,end));
    end
    F(1) = P/2;

    % the rotation of the beam segments is equal to zero, w_k' = 0
    for x = 1:n+1
        y = y+1;
        A(y,4*(interface(x,end)-1)+2:4*interface(x,end)) =...
            [1, L, 3*(L/2)^2];
    end

    % the displacements of the beam segements are the same at the end of the
    % clamped, w_k = w_k+1
    for x = 1:n
        y = y + 1;
        A(y,4*(interface(x,end)-1)+1:4*interface(x,end)) = ...
            [1, L/2, (L/2)^2, (L/2)^3];
        A(y,4*(interface(x+1,end)-1)+1:4*interface(x+1,end)) =...
            -[1, L/2, (L/2)^2, (L/2)^3];
    end

case 2
    % boundary formulations for the cantilevered beam
    % The conditions at the root of the beam (x=0) does not need to be taken into
    % acount since it would yield in two rows where all entries are equal to zero

    % at the end of the cantilever, x = L
    % applied force P equals the sum of the shear forces V_k acting in the
    % segments. V_k = M_k' = -EI_k*w_k''' = -EI_k(6*C_k,4)
    for x = 1:n+1
        A(y,interface(x,end)*4) = -6*EI(interface(x,end));
```

```
    end
    F(1) = P;

    % the internal moment in each segment is zero, M_k = 0
    for x = 1:n+1
      y = y + 1;
      A(y,4*(interface(x,end)-1)+3:4*interface(x,end)) = ...
        [-2*E*I(interface(x,end)) -6*EI(interface(x,end))*L];
    end

    % the displacements of the beam segements are the same at the end of the
    % cantilever, w_k = w_k+1
    for x = 1:n
      y = y + 1;
      A(y,4*(interface(x,end)-1)+1:4*interface(x,end)) = [1, L, L^2, L^3];
      A(y,4*(interface(x+1,end)-1)+1:4*interface(x+1,end)) =...
        -[1, L, L^2, L^3];
    end

end

%% -------------------------- continuity equations ---------------------------- %%

% at each delamination tip we have to see how many segments are joined up and set up
% the equations accordingly. We always have though that the displacement of the
% "intact" beam is equal to each of the resulting segments and equally for the
% rotation. Further are both shear force and moment in the intact segment equal to
% the sum of shear forces and moments in the resulting segments.
% w_j = w_k, w_j = w_l, w_j' = w_k', w_j' = w_l', M_j = M_k + M_l, V_j = V_k + V_l

% go through each crack
for x = 1:size(interface,2)
  % choose the appropriate elements for the defintion of the crack tip equations
  if x == 1
    % "intact" segment
    j = elems(end);
    % "split" segements
    split{1} = unique(interface(:,1));
  else
    % determine those locations where a delamination is present
    temp = interface(:,x) - interface(:,x-1);
    % find non-zero elements, as they have been split
    temp1 = find(temp);
    % use those indices in interface(:,x-1) to see which element has been split
    temp2 = interface(temp1,x-1);
    temp3 = unique(temp2);
    if length(temp3) == 1
      % "intact" segment
      j = temp3;
      % "split" segements
      split{1} = unique(interface(temp1,x));
    else
      % "intact" segments, this will set up another loop below
      j = temp3;
      for tl = 1:length(temp3)
        temp4 = find(temp2 == temp3(tl));
        split{tl} = unique(interface(temp1(temp4),x));
      end
```

```
  end
end


% loop through j in case there are several splits at different heights
for jl = 1:length(j)

  %-------------------- displacement j - x ----------------------------------%
  % depending on the size of split we have more than two segements split from
  % segment j
  for sl = 1:size(split{jl},1)
    k = split{jl}(sl);
    y = y + 1;
    if j(jl) == elems(end)
      if BC == 0 % adust the expression for w_0
        A(y,4*(j(jl)-1)+1:4*(j(jl)-1)+2) = [xc(x) xc(x)^3];
      else
        A(y,4*(j(jl)-1)+1:4*(j(jl)-1)+2) = [xc(x)^2 xc(x)^3];
      end
    else
      A(y,4*(j(jl)-1)+1:4*j(jl)) = [1 xc(x) xc(x)^2 xc(x)^3];
    end
    A(y,4*(k-1)+1:4*k) = -[1 xc(x) xc(x)^2 xc(x)^3];
  end

  %-------------------- rotation j - x --------------------------------------%
  % see displacemnt for comment
  for sl = 1:size(split{jl},1)
    k = split{jl}(sl);
    y = y + 1;
    if j(jl) == elems(end)
      if BC == 0 % adust the expression for w_0
        A(y,4*(j(jl)-1)+1:4*(j(jl)-1)+2) = [1 3*xc(x)^2];
      else
        A(y,4*(j(jl)-1)+1:4*(j(jl)-1)+2) = [2*xc(x) 3*xc(x)^2];
      end
    else
      A(y,4*(j(jl)-1)+2:4*j(jl)) = [1 2*xc(x) 3*xc(x)^2];
    end
    A(y,4*(k-1)+2:4*k) = -[1 2*xc(x) 3*xc(x)^2];
  end

  %-------------------- Moment equilibrium ----------------------------------%
  % depending on how many segments are split we have to take the sum of all of
  % them
  y = y + 1;
  if j(jl) == elems(end)
    if BC == 0
      A(y,4*(j(jl)-1)+2) = -6*EI(j(jl))*xc(x);
    else
      A(y,4*(j(jl)-1)+1:4*(j(jl)-1)+2)=-[2*EI(j(jl)) 6*EI(j(jl))*xc(x)];
    end
  else
    A(y,4*(j(jl)-1)+3:4*j(jl)) = -[2*EI(j(jl)) 6*EI(j(jl))*xc(x)];
```

```
    end
    for sl = 1:size(split{jl},1)
      k = split{jl}(sl);
      A(y,4*(k-1)+3:4*k) = [2*EI(k) 6*EI(k)*xc(x)];
    end

    %-------------------- Force equilibrium ----------------------------------%
    % see moment equilibrium for comment
    y = y + 1;
    if j(jl) == elems(end)
      A(y(jl),4*(j(jl)-1)+2) = -6*EI(j(jl));
    else
      A(y,4*j(jl)) = -6*EI(j(jl));
    end
    for sl = 1:size(split{jl},1)
      k = split{jl}(sl);
      A(y,4*k) = 6*EI(k);
    end
  end
end

%% -------------------------------- solve the system --------------------------- %%
% SVD approach
tol = size(A,2)*2^-53;
[U,S,V] = svd(A);
Sinv = zeros(size(A,2));
for il = 1:size(A,2)
  if (S(il,il)/S(1) < tol) | (S(il,il) == 0)
    Sinv(il,il) = 0;
  else
    Sinv(il,il) = 1/S(il,il);
  end
end

if size(A,1) == size(A,2)
  % solution vector according to SVD
  C = V*Sinv*U'*F;
else
  C = 0;
  for il = 1:size(A,2)
    C = C + dot(U(:,il),F)*Sinv(il,il)*V(:,il);
  end
end

%% rearranging the coefficients, making them easier to discern
Cmat = zeros(4,elemcount);
for ind = 1:elemcount
  if ind == elemcount
    if BC == 0
      Cmat(:,ind) = [0; C(4*(ind-1)+1); 0; C(4*(ind-1)+2)];
    else
      Cmat(:,ind) = [0;0; C(4*(ind-1)+1:4*(ind-1)+2)];
    end
  else
    Cmat(:,ind) = C(4*(ind-1)+1:4*ind);
  end
```

```
end

end


%% ------------------------- ERR_strain.m ------------------------------------ %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% this function has as only purpose the computation of the ERR for a delamination tip
% based on the approach published by Andrews. To that end it accepts a matrix (tips)
% and an indicator (indv) to alter a specific row in the matrix by adding DELa to it.
% In an iterative process this delamination is brought to its original length and the
% final result for the ERR is returned back to the calling function.
% ATTENTION: NOW WE WILL ONLY EVER CHANGE JUST ONE DELAMINATION AT A TIME!
% ------------------------------------------------------------------------------ %
% Input:
% - tips0 (matrix containing delamination length and position within the stack
% - indv (points towards the entry of tips0 that needs to be altered
% - contact (contact selection: 0 -> unconstrained; 1-> constrained)
% - P (current applied force)
% ------------------------------------------------------------------------------ %
% Output:
% - G, Energy Release Rate for the tip under consideration
% - SIF, Stress Intensity factor
%------------------------------------------------------------------------------%
% - author: J.J. Kurpierz
% - date: 16.10.2012
% - version: V3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of function
function [G,SIF] = ERR_strainV5(tips0, indv, contact, P)
%% Variable declaration
global tar

%% input
% initial delamination extension. If this method is applied to all equal length
% delaminations then simply add 4E0(5). Otherwise, take half the difference wrt the
% next larger delamination
err = 1;
co = 1;
% if tips0(indv,1) == max(tips0(:,1))
  da(co) = 2*tar.E0(7);
% else
%     % find next larger delamination size
%     temp = sort(unique(tips0(:,1)), 'descend');
%     % compute the initial extension length
%     da(co) = (temp(find(temp == tips0(indv,1))-1) - tips0(indv,1))/2
%     if da(co) < 1e-6
%
% end

while err > 1e-6 && da(co) > 1e-6
  % extend the delaminations in question by da
  tips = tips0;
  tips(indv,1) = tips0(indv,1) + da(co);

  % call on numbering_func in order to obtain the interface matrix, sorted tips,
  % height vector and the number of elements

  [interface, tipsn, seqk, height, elemcount] = interface_funcV3(tips);
```

```
seqk{elemcount} = tar.seq;
height(elemcount) = tar.h;

   %% segment and beam properties
% get the separate seqk and compute the D-matrix (reduced version) for it
for el = 1:elemcount
   [K, ~] = ABDmat(tar.E0, seqk{el}, 0, 0);
   % reduced bending matrix D due to unsymmetric and unbalanced laminates
   Dred = K(4:6,4:6);
   % compliance matrix
   dred = inv(Dred);
   % Youngs modulus of the composite plate, based on bending stiffness (see
   % Kassapoglou or Reddy)
   EI(el) = tar.w/dred(1,1);
end

% section flexural rigidity for constrained contact
n = size(interface,2) + 1;
for x = 1:n-1
  EIsec(x) = sum(EI(unique(interface(:,x))));
end
EIsec(n) = EI(end);

% tip location
xc = tar.L/2-flipud(unique(tipsn(:,1)));

   %% solve the system of equations
% use "contact" as indicator to chose from the rigth function
if contact == 0
   Cmat = SoE_ucbV2(interface,EI,xc,P);
else
   Cmat = SoE_cbV2(interface,EIsec,xc,P);
end

   %% Compute ERR based on the Strain Energy
% integration boundary
intbound = [0;xc;tar.L/2];

% switch between the two contact cases
switch contact
   case 0
     % integration
     for pl = 1:n
       % first part involving the undamaged beam segment
       if pl == 1
          Up(pl) = EI(end)*...
                (4*Cmat(3,end)^2*(intbound(pl+1)-intbound(pl))+...
                 12*Cmat(3,end)*Cmat(4,end)*(intbound(pl+1)^2-intbound(pl)^2)+...
                 12*Cmat(4,end)^2*(intbound(pl+1)^3-intbound(pl)^3));
       else
          segs = unique(interface(:,pl-1));
          Upi = [];
          for sl = 1:length(segs)
             Upi(sl) = EI(segs(sl))*...
                (4*Cmat(3,segs(sl))^2*(intbound(pl+1)-intbound(pl))+...
                 12*Cmat(3,segs(sl))*Cmat(4,segs(sl))*...
                 (intbound(pl+1)^2-intbound(pl)^2)+...
```

```
                12*Cmat(4,segs(sl))^2*(intbound(pl+1)^3-intbound(pl)^3));
            end
            Up(pl) = sum(Upi);
        end
    end

  case 1
    for pl = 1:n
      if pl == 1
        Up(pl) = EIsec(end)*...
        (4*Cmat(3,end)^2*(intbound(pl+1)-intbound(pl))+...
        12*Cmat(3,end)*Cmat(4,end)*(intbound(pl+1)^2-intbound(pl)^2)+...
        12*Cmat(4,end)^2*(intbound(pl+1)^3-intbound(pl)^3));
      else
        sec = pl-1;
        Up(pl) = EIsec(sec)*...
        (4*Cmat(3,sec)^2*(intbound(pl+1)-intbound(pl))+...
        12*Cmat(3,sec)*Cmat(4,sec)*(intbound(pl+1)^2-intbound(pl)^2)+...
        12*Cmat(4,sec)^2*(intbound(pl+1)^3-intbound(pl)^3));
      end
    end

  end

  % total Strain Energy
  U(co) = 1/2*sum(Up);

  % compute the gradient based on a forward difference
  if co > 2
    dUda_old = (U(co-2)-U(co-1))/((da(co-2)-da(co-1))*tar.w);
    dUda_new = (U(co-1)-U(co))/((da(co-1)-da(co))*tar.w);
    err = abs((dUda_new-dUda_old)/dUda_old);
  end
  co = co + 1;
  da(co) = 0.5*da(co-1);

end
G = dUda_new;

%% compute the interface SIF for only the current delamination tip
% first the "right" tip elements have to be chosen. To this end I need the
% respective column of the interface matrix. I get that by looking at the order
% of the current delamination within tips
section = find(sort(unique(tips(:,1)), 'descend') == tips(indv,1));
% According to my thinking all I have to do now is to subtract column "order-1"
% from column "order" and look at the non-zero items. THey indicate the newly
% created longer delamination under consideration. Finally, I simply need the
% element that those two are connected to.
% keyboard
if section == 1
  temp = interface(:,section);
else
  temp = interface(:,section)-interface(:,section-1);
end
% filter out the only two unique elements right of the crack
```

```
temp1 = unique(interface(find(temp),section));
seg.k = temp1(1);
seg.l = temp1(2);
sec.kl = section;
% find the element that both of these elements are connected to
if section == 1
   seg.j = elemcount;
   sec.j = n;
else
   seg.j = unique(interface(find(temp),section-1));
   sec.j = section-1;
end
% keyboard
SIF = SIF_funcV3(seg,sec,Cmat,height,seqk,xc(section));

end


%% --------------------------- delam_growth_appdisp.m ----------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function for determining the growth of a delamination under applied displacement.
% This function will be called as soon as growth has been determined in any of the
% delaminations.
%-------------------------------------------------------------------------------------%
% input:
% - tips,
% - indv,
% - contact,
% - P
% - all input is given in the structures "dam" and "tar"
%-------------------------------------------------------------------------------------%
% output:
% - new applied load P, Pnew
% - new Energy Release Rate G
% - updated dam.delams
%-------------------------------------------------------------------------------------%
% info:
% - author: J.J. Kurpierz
% - date: 14.11.2012
% - Version: V1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of fucntion
function[Pnew, Gnew, SIFnew] = delam_growth_appdispV2(indv,contact,wc)

%% Variable declaration
global dam tar

%% change in delamination length
aext = tar.E0(7);
Gnew = dam.G(indv,1);
intnum = round(dam.delams(indv,2)/tar.E0(7));
while Gnew > tar.Gcr
  % let the delamination in question grow
  fprintf(1,'delamination at interface %i grows by one ply thickness\n', intnum)
  dam.delams(indv,1) = dam.delams(indv,1)+aext;

  % call interface_func to determine new interface
  [interface, tips, seqk, height, elemcount] = interface_funcV3(dam.delams);
  seqk{elemcount} = tar.seq;
```

```
   height(elemcount) = tar.h;

      %% segment and beam properties
   % for each beam segement the reduced bending stiffness matrix needs to be computed.
   for el = 1:elemcount
      [K, ~] = ABDmat(tar.E0, seqk{el}, 0, 0);
      % reduced bending matrix D due to unsymmetric and unbalanced laminates
      Dred = K(4:6,4:6);
      % compliance matrix
      dred = inv(Dred);
      % Youngs modulus of the composite plate, based on bending stiffness (see
      % Kassapoglou or Reddy)
   %          E(el) = 12/(height(el)^3*dred(1,1));
      EI(el) = tar.w/dred(1,1);
   end

   % section flexural rigidity for constrained contact
   n = size(interface,2) + 1;
   for x = 1:n-1
     EIsec(x) = sum(EI(unique(interface(:,x))));
   end
   EIsec(n) = EI(end);

   % x-coordinate for the crack tips
   xc = tar.L/2-flipud(unique(tips(:,1)));

   % solve the system of equations resulting from the unknown segment and section
   % displacements, based on the chosen contact formulation.
   % This will serve as the new compliance (see page 31 in NB 3). Use unit load as input
   if contact == 0
      Cmat = SoE_ucbV2(interface,EI,xc,1);
   else
      Cmat = SoE_cbV2(interface,EIsec,xc,1);
   end

   % displacement due to unit load
   wc_u = Cmat(1,end-1) + Cmat(2,end-1)*tar.L/2 +...
         Cmat(3,end-1)*(tar.L/2)^2 + Cmat(4,end-1)*(tar.L/2)^3;

   % new applied load
   Pnew = wc/wc_u;

      %% check with Pnew and new length for G
   [Gnew, SIFnew] = ERR_strainV5(dam.delams, indv, contact, Pnew);
end

end
```

```
%% ------------------------- SIF_func.m ---------------------------------------- %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function computes the Stress Intensity Factor based on the expressions derived
% by Suo & Hutchinson.
%-------------------------------------------------------------------------------%
% input:
% - j,k,l: indices for the three segments that make up the crack-tip element
% - Cmat, matrix containing the displacement function coefficients
% - seqk, cell containing the sublaminate layups for the definition of the bimaterial
% constant
% - height, vector containing heights of the segments
%-------------------------------------------------------------------------------%
% output:
% - SIF, vector containing KI and KII and bi-material constant epsilon
%-------------------------------------------------------------------------------%
% author: J.J. Kurpierz
% Version: V2
% date: 15.11.2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% start of the function
function[SIF] = SIF_funcV3(seg,sec,Cmat,height,seqk,xa)

%% Variable declaration
global tar sim

%% preliminaries
% define all the parameters needed based on the information about the
% layer j
[~, EngCst] = ABDmat(tar.E0,seqk{seg.j},0,0);
E_0 = EngCst(2,1);

% layer k
[~, EngCst] = ABDmat(tar.E0,seqk{seg.k},0,0);
E1 = EngCst(2,1);
mu1 = EngCst(3,2);
nu1 = EngCst(5,1);
% layer l
[~, EngCst] = ABDmat(tar.E0,seqk{seg.l},0,0);
E2 = EngCst(2,1);
mu2 = EngCst(3,2);
nu2 = EngCst(5,1);

Gam = mu1/mu2;
kap1 = (3 - nu1)/(1+nu1);    % plane stress
kap2 = (3 - nu2)/(1+nu2);    % plane stress
alpha = (Gam*(kap2+1)-(kap1+1))/(Gam*(kap2+1)+(kap1+1));
beta = (Gam*(kap2-1)-(kap1-1))/(Gam*(kap2+1)+(kap1+1));
c1 = (kap1 + 1)/mu1;
c2 = (kap2 + 1)/mu2;
SIG = c2/c1;
p = sqrt((1-alpha)/(1-beta^2));
epsi = 1/(2*pi)*log((1-beta)/(1+beta));

% define the geometry of the crack element
h(1) = height(seg.k);
h(2) = height(seg.l);
I1 = 1/12*h(1)^3*tar.w;
I2 = 1/12*h(2)^3*tar.w;
```

```
I_0 = 1/12*height(seg.j)^3*tar.w;

if h(1) > h(2)
   eta = h(2)/h(1);
   h1d = h(2);
else
   eta = h(1)/h(2);
   h1d = h(1);
end

% more paramters
A = 1/(1+SIG*(4*eta + 6*eta^2 + 3*eta^3));
I = 1/(12*(1+SIG*eta^3));
gam = rad2deg(asin(6*SIG*eta^2*(1+eta)*sqrt(A*I)));
omega = 52.1-3*eta; % in degrees

% definition of Ci, i = 2,3 from Appendix III (Suo&Hutchinson)
DEL = (1+2*SIG*eta+SIG*eta^2)/(2*eta*(1+SIG*eta));
I0 = 1/3*(SIG*(3*(DEL-1/eta)^2 - 3*(DEL-1/eta)+1) +...
      3*DEL/eta*(DEL-1/eta) + 1/eta^3);
C2 = SIG/I0*(1/eta - DEL + 1/2);
C3 = SIG/(12*I0);

%% Definition of SIF based on crack-tip resultant stresses
% crack tip stress resultants
if sim.contact == 0

   M0 = -E_0*I_0*(2*Cmat(3,seg.j)+6*Cmat(4,seg.j)*xa)/tar.w;
   if h(1) > h(2)
      M1 = -E2*I2*(2*Cmat(3,seg.l)+6*Cmat(4,seg.l)*xa)/tar.w;
   else
      M1 = -E1*I1*(2*Cmat(3,seg.k)+6*Cmat(4,seg.k)*xa)/tar.w;
   end
else
   M0 = -E_0*I_0*(2*Cmat(3,sec.j)+6*Cmat(4,sec.j)*xa)/tar.w;
   if h(1) > h(2)
      M1 = -E2*I2*(2*Cmat(3,sec.kl)+6*Cmat(4,sec.kl)*xa)/tar.w;
   else
      M1 = -E1*I1*(2*Cmat(3,sec.kl)+6*Cmat(4,sec.kl)*xa)/tar.w;
   end
end

N_1 = -C2*M0/h1d;
M_1 = M1 - C3*M0;

KII = p/sqrt(2)*(N_1/sqrt(A*h1d)*sind(omega)-...
   M_1/sqrt(I*h1d^3)*cosd(gam+omega));
KI = p/sqrt(2)*(N_1/sqrt(A*h1d)*cosd(omega)+...
   M_1/sqrt(I*h1d^3)*sind(gam+omega));

K = (KI+KII*1i)*h1d^(-1i*epsi);
G_k = (c1+c2)/(16*cosh(pi*epsi)^2)*abs(K)^2;

SIF = [KI KII epsi h1d G_k];
```

```
end
```