

# Master's Thesis Master in Aeronautical Engineering

## Prediction of crack onset and propagation in adhesive joints by means of the Coupled Criterion of the Finite Fracture Mechanics

Autor: Andrés Laguna Alcalde

Tutor: María del Mar Muñoz-Reja Moreno

Luis Arístides Távara Mendoza

**Dpto. Elasticidad y Resistencia de Materiales  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla**

Sevilla, 2023





Master's Thesis  
Master in Aeronautical Engineering

**Prediction of crack onset and propagation in  
adhesive joints by means of the Coupled Criterion  
of the Finite Fracture Mechanics**

Autor:

Andrés Laguna Alcalde

Tutor:

María del Mar Muñoz-Reja Moreno  
Luis Arístides Távara Mendoza

Profesor Contratado Doctor  
Profesor Titular

Dpto. Elasticidad y Resistencia de Materiales  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023





# Acknowledgements

---

To my family.  
To my friends.  
To professors who believe in students and education.

*Andrés Laguna Alcalde*  
*Sevilla, 2023*



# Resumen

---

Últimamente, el Criterio Acoplado de la Mecánica de la Fractura Finita (CC-FFM) está siendo ampliamente usado para precedir el inicio y propagación de grietas en gran variedad de campos de aplicación. En este proyecto, la formulación específica basada en la Minimización de la Energía Total (conformada por potencial más liberada) sujeta a Condición de Tensión (PMTE-SC) será aplicada numéricamente a uniones del tipo Doble Viga en Voladizo (DCB) y Doble Unión a Solape (DLJ). Se empleará ingeniería inversa para caracterizar las propiedades de Interfases Frágiles Elastolineales (LEBIM) a través de experimentos de la literatura. Se compararán resultados numéricos y analíticos para validar la herramienta Python-ABAQUS en la que el proyecto se ha basado.



# Abstract

---

The Coupled Criterion of Finite Fracture Mechanics (CC-FFM) is currently being highly used to predict crack onset and propagation in many fields. In this project, the particular formulation based on the Minimization of Total Energy (composed of potential plus released energy) subjected to Stress Condition (PMTE-SC) will be numerically applied to Double Cantilever Beams (DCB) and Double Lap Joints (DLJ). Reverse engineering will be employed to characterize Linear Elastic Brittle Interfaces (LEBIM) properties through experiments from the literature. Comparisons between numerical and analytical results will be made to validate the Python-ABAQUS tool in which the project is based.



# Notation

---

<i>AMA</i>	Alternating Minimization Algorithm
<i>BC</i>	Boundary Conditions
<i>CCFFM</i>	Couple criterion of Finite Fracture Mechanics
<i>CFRP</i>	Carbon Fiber Reinforced Polymeric
<i>CZM</i>	Cohesive Zone Model
<i>DCB</i>	Double Cantilever Beam
<i>DLJ</i>	Double Lap Joint
<i>ERR</i>	Energy Release Rate
<i>FEM</i>	Finite Element Method
<i>FFM</i>	Finite Fracture Mechanics
<i>LEBIM</i>	Linear Elastic Brittle Interface Model
<i>PMTE – SC</i>	Principle of Minimum Total Energy subjected to Stress Condition
<i>SLJ</i>	Single Lap Joint
<i>UMAT</i>	User-Material
<i>US</i>	University of Seville





# Abbreviated Index

---

<i>Acknowledgements</i>	I
<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notation</i>	VII
<i>Abbreviated Index</i>	IX
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and objectives	1
1.2 Structure of the present work	1
<b>2 Study of the art and theoretical framework</b>	<b>3</b>
2.1 Linear elastic fracture mechanics (LEFM)	3
2.2 Cohesive zone models (CZM)	3
2.3 Linear elastic-brittle interface model (LEBIM)	4
2.4 Finite fracture mechanics (FFM)	7
<b>3 Numerical implementation</b>	<b>11</b>
3.1 Nomenclature and minimization subsets	11
3.2 Iterations	13
3.3 Algorithm process	14
3.4 DCB example	15
<b>4 Tests: Double Cantilever Beam (DCB)</b>	<b>21</b>
4.1 Problem description	21
4.2 Effect of the applied load location	22
4.3 Parameter study: step increment	23
4.4 Mesh convergence study	25
4.5 Accuracy rising by code development	26
4.6 Energetic understanding of PMTE-SC applied to DCB	33
<b>5 Tests: Double Lap Joint (DLJ)</b>	<b>37</b>
5.1 Problem description	37
5.2 Comparison between analytical and numerical models	37
5.3 PMTE-SC. Adhesive characterization	43

<b>6 Unification of codes</b>	<b>67</b>
<b>7 Conclusions</b>	<b>69</b>
7.1 Future Work	69
<b>Appendix A Scripts to launch parameter loops</b>	<b>71</b>
<b>Appendix B Create an executable to run python scripts</b>	<b>75</b>
<b>Appendix C How to prepare and run PMTE-SC codes</b>	<b>79</b>
C.1 Launch multiprocessing from Python 2.7	79
C.2 Launch from ABAQUS (Vers. 6.14)	84
<i>List of Figures</i>	87
<i>List of Tables</i>	89
<i>Bibliography</i>	91

# Contents

---

<i>Acknowledgements</i>	I
<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notation</i>	VII
<i>Abbreviated Index</i>	IX
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and objectives	1
1.2 Structure of the present work	1
<b>2 Study of the art and theoretical framework</b>	<b>3</b>
2.1 Linear elastic fracture mechanics (LEFM)	3
2.2 Cohesive zone models (CZM)	3
2.3 Linear elastic-brittle interface model (LEBIM)	4
2.3.1 Constitutive law of the spring distribution	4
2.3.2 Interface failure criterion	5
2.4 Finite fracture mechanics (FFM)	7
2.4.1 Coupled criterion of finite fracture mechanics (CCFFM) by means of LEBIM formulation	7
2.4.2 Principle of minimum total energy subjected to stress condition (PMTE-SC)	9
<b>3 Numerical implementation</b>	<b>11</b>
3.1 Nomenclature and minimization subsets	11
3.2 Iterations	13
3.3 Algorithm process	14
3.4 DCB example	15
3.4.1 Displacement control	15
3.4.2 Load control	18
<b>4 Tests: Double Cantilever Beam (DCB)</b>	<b>21</b>
4.1 Problem description	21
4.2 Effect of the applied load location	22
4.3 Parameter study: step increment	23
4.4 Mesh convergence study	25
4.5 Accuracy rising by code development	26
4.5.1 High accuracy model with fixed initial increment	27
4.5.2 Final improvements to rise computational efficiency of the codes	28

4.6	Energetic understanding of PMTE-SC applied to DCB	33
4.6.1	Displacement control	34
4.6.2	Load control	35
<b>5</b>	<b>Tests: Double Lap Joint (DLJ)</b>	<b>37</b>
5.1	Problem description	37
5.2	Comparison between analytical and numerical models	37
5.2.1	Isotropic adherends	39
5.2.2	Combination of orthotropic and isotropic adherends	41
5.3	PMTE-SC. Adhesive characterization	43
5.3.1	Determination of $k_t$	44
5.3.2	Determination of $G_{II}^{cr}$ and $\tau_{cr}$ . Reverse engineering	48
	Surfaces	56
	Intersection curves	58
5.3.3	Curves of energetic behaviour	63
	Load control	64
	Displacement control	66
<b>6</b>	<b>Unification of codes</b>	<b>67</b>
<b>7</b>	<b>Conclusions</b>	<b>69</b>
7.1	Future Work	69
	<b>Appendix A Scripts to launch parameter loops</b>	<b>71</b>
	<b>Appendix B Create an executable to run python scripts</b>	<b>75</b>
	<b>Appendix C How to prepare and run PMTE-SC codes</b>	<b>79</b>
C.1	Launch multiprocessing from Python 2.7	79
C.1.1	Procedure to launch the codes	79
C.1.2	Procedure to prepare the codes	81
C.2	Launch from ABAQUS (Vers. 6.14)	84
C.2.1	Procedure to launch the codes	84
C.2.2	Procedure to prepare the codes	84
	<i>List of Figures</i>	87
	<i>List of Tables</i>	89
	<i>Bibliography</i>	91

# 1 Introduction

---

*Everything is theoretically impossible, until it is done.*

ROBERT A. HEINLEIN

First of all, we will discuss the motivation and objectives of the present work, as well as expose its structure.

## 1.1 Motivation and objectives

During the last century, much researches have been conducted on the different types of joints that can be used for assemblies, since they are critical areas of structures. In fact, both their properties and weight make a difference in the behavior of the structure, as well as in operating and maintenance costs. Some of the most commonly used joints are welding, bolted joints, and adhesive joints. In this project, we aim to focus on the last ones.

Several methods have been developed for the study of adhesive joints. Cohesive models are one of the most widely used in the industry, although they have the disadvantage of requiring high computational costs. Under the interest of obtaining more efficient methods, studies emerged within the field of fracture mechanics such as LEBIM, a model that will be explained in this project and implemented through the Principle of Minimum Total Energy Subjected to Stress Condition (PMTE-SC) in the finite element numerical tool ABAQUS to predict crack onset and propagation.

The objective of the present work is to understand, improve and validate the numerical tool developed by the Department of Elasticity and Resistance of Materials of the University of Seville. This tool consists in the interaction of Python scripts and ABAQUS models whose aim is to predict the crack onset and propagation at the adhesive which bonds a specimen.

## 1.2 Structure of the present work

We will start by presenting the context and theoretical framework (chapter 2) in which the research is conducted.

Once the basis are understood, the gross of the project will be developed in chapter 4 and chapter 5, where two of the most typical joints (Double Cantilever Beam and Double Lap Joint) will be analysed and tested.

To complete the project and to clarify the script structure and functionality, in chapter 6, parallel developments of the codes will be unify.

Finally, main conclusions derived from the development and composition of the project will be addressed in chapter 7.



## 2 Study of the art and theoretical framework

---

*You must be the change you wish to see in the world.*

MAHATMA GHANDI

In this section, a brief introduction about the state of the art of the context in which the project is developed will be made. In this line, the main authors involved in the development of fracture mechanics will be mentioned and more focus will be made around the concepts and models involved in this project.

### 2.1 Linear elastic fracture mechanics (LEFM)

The crack onset and propagation through different materials and configurations has been constantly studied in the last century. With this necessity, the Fracture Mechanics Field was born.

Griffith made the first steps around this topic ([1]), proposing a simple energetic balance to justify crack growths. He proposed that the energy required to open new surfaces in the material derived from the elastic potential energy stored in the material and that it was proportional to the energy required to break atom bonds, involving infinitesimal crack growths. This way, the energy release rate (ERR) was firstly defined as the decrement of the potential energy per unit area. However, this energetic approach did not consider phenomena like plastification, where the order of magnitude of energy is considerably bigger compared to breakages of atom bonds.

For this reason, other formulations arrived later. Irwin ([6]) gave a stress perspective to this issue obtaining an asymptotic solution for the stress and displacement field ( $\sigma_{ij} \propto 1/\sqrt{2\pi r}$ ).

The application of Griffith's criterion to a semi-infinite plate under tension with a preexisting crack of length  $2a$  establishes that it would propagate when a critical value (proportional to  $1/\sqrt{a}$ ) is reached.

As it is observed, both approaches are only possible under the premise of an existing crack so they achieved to predict propagation, but not onsets.

### 2.2 Cohesive zone models (CZM)

During the 60's, Cohesive Zone Models (CZM) were developed to model brittle and ductile fractures. It was mainly applied to concrete, but it has also been used in composite materials lately. These models are based on Griffith's theory but they remove the infinite jump of stress along the crack path to allow its propagation. This way, crack onsets can be predicted.

CZM are typically used due to their accuracy. However, the required computational cost is incredibly high. For this reason, researchers keep formulating new models, such as phase field models, which are not addressed in this project, or LEBIM, which can be considered as a limit case of cohesive models.

## 2.3 Linear elastic-brittle interface model (LEBIM)

Prandtl ([17]) proposed a linear distribution of elastic springs to model brittle interface fractures where adherends stiffness dominate, typically known as linear elastic-brittle interface model (LEBIM). This concept was later employed by many authors to develop different models that allows to study not only the crack propagation, but also crack onset. In particular, new constitutive laws were implemented for fiber-matrix interfaces in [20] under transverse uniaxial load and in [23] and [22] under biaxial loads.

The LEBIM has been employed in Boundary Element Models (BEM) and FEM (Finite Element Models) and its application might help to study the fiber-matrix interface, where failure under transverse (to fiber) loads is usually provoked by the debonding of some fibers from the matrix. This way, it is believed that this interface could be modelled by the LEBIM, although strictly speaking, there is no intermediate material between fiber and matrix. In fact, it is in our project where the adhesives will be approximated as interfaces.

### 2.3.1 Constitutive law of the spring distribution

In this section, the constitutive law of the LEBIM will be attempted. This model simplifies interfaces as thin linear-elastic layers between two surfaces. The undamaged layer, which is considered a solid of Young's modulus  $E$ , Poisson ratio  $\nu$ , shear modulus  $\mu$ , Lamé's parameter  $\lambda$ , width  $w$  and small thickness  $h$ , is modelled as a continuous spring distribution just like Figure 2.1 shows.

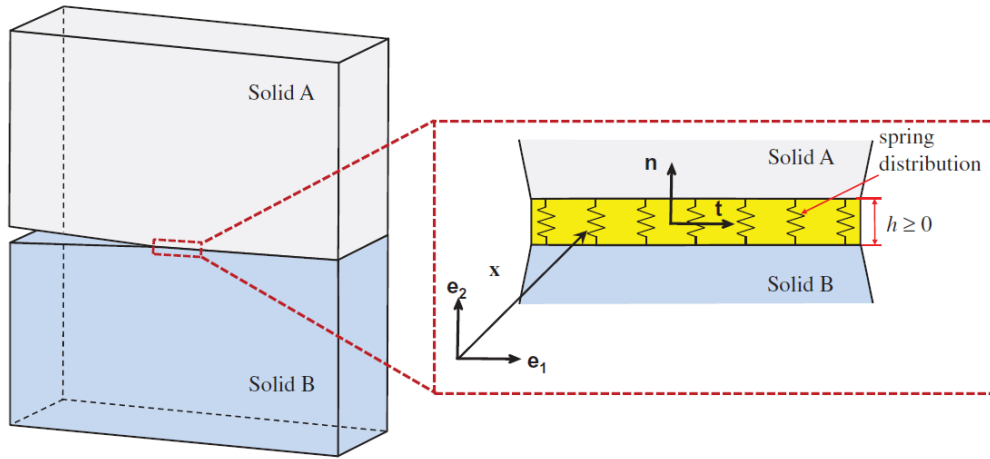


Figure 2.1 Interface modelled through spring distribution [22].

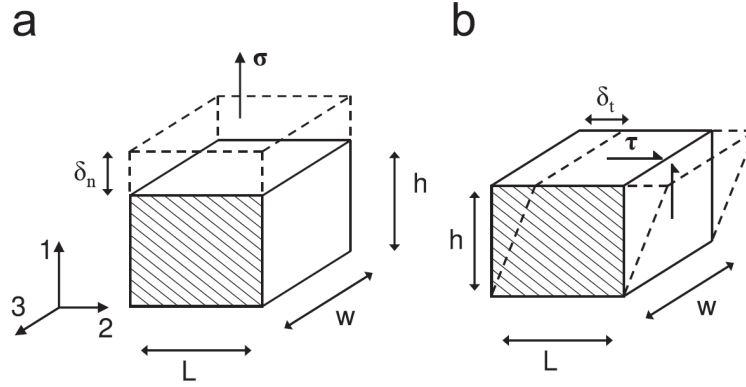
The stiffness parameters of the springs can be written in terms of the just detailed isotropic parameters of the layer. By starting from stress-displacement laws ( $\sigma_{11} = E\varepsilon_{11} = E\frac{\delta_n}{h}$  and  $\sigma_{12} = 2\mu\varepsilon_{12} = \mu\frac{\delta_t}{h}$ ), where  $\sigma_{11}$  and  $\sigma_{12}$  are normal and shear stresses, respectively;  $\varepsilon_{11}$  and  $\varepsilon_{12}$  are normal and tangential strains, respectively;  $\delta_n$  and  $\delta_t$  are the normal and tangential displacements, respectively, between top and bottom surfaces of the elastic layer (see Figure 2.2). Therefore, if  $\sigma = k_n\delta_n$  and  $\tau = k_t\delta_t$ , thus

$$\begin{cases} k_n = \frac{E}{h} \\ k_t = \frac{\mu}{h} \end{cases} \quad (2.1)$$



It is assumed that if the adhesive layer is thin and considerably less stiff than the solids, the expansions due to Poisson effect are null under loads in plane 1,2 (see axes in Figure 2.2). Therefore, normal strains in directions perpendicular to 3 are null:  $\varepsilon_{22} = \varepsilon_{33} = 0$ . Then, just like [20] demonstrates, the relation between normal and tangential adhesive stiffness (Equation 2.2) can be obtained, where it can be seen that  $2 \leq \frac{k_n}{k_t} < \infty$ .

$$\frac{k_n}{k_t} = \frac{2\mu + \lambda}{\mu} = \frac{2}{1 - \nu'} = \frac{2(1 - \nu)}{1 - 2\nu} \quad (2.2)$$



**Figure 2.2** Portion of the linear-elastic layer under normal (a) and tangential (b) stresses [20].

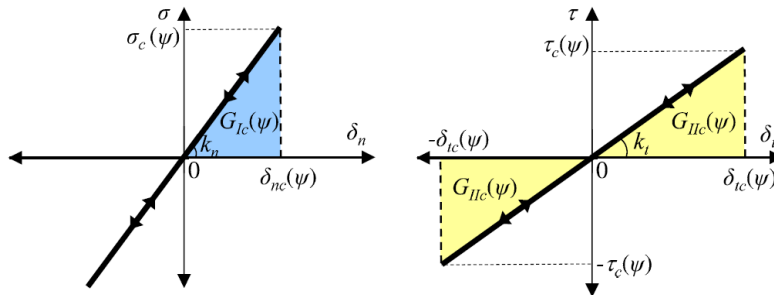
### 2.3.2 Interface failure criterion

The model proposes a linear elastic-brittle law for the elastic layer in 2D written at an interface point  $x$ , where the spring is located.

The constitutive laws for an undamaged and broken interfaces are written in Equation 2.3 and Equation 2.4, respectively, and represented in Figure 2.3 and Figure 2.4, respectively, for both normal and tangential stresses.

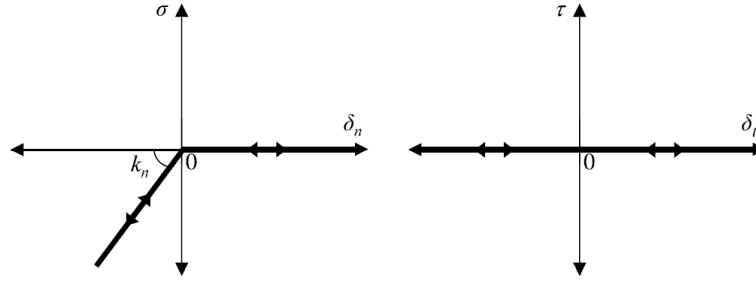
$$\text{Undamaged interface} \begin{cases} \sigma(x) = k_n \delta_n(x) \\ \tau(x) = k_t \delta_t(x) \end{cases} \quad \text{for } G(x) < G_c(\psi(x)) \quad (2.3)$$

$$\text{Broken interface} \begin{cases} \sigma(x) = k_n \langle \delta_n(x) \rangle_- \\ \tau(x) = 0 \end{cases} \quad (2.4)$$



**Figure 2.3** Normal (a) and tangential (b) linear elastic-brittle law for undamaged interface [23].

What these laws model is that when the spring is undamaged, it might be loaded normal or tangentially; whereas when it is broken it only withstands compression loads. This is done because it seems reasonable



**Figure 2.4** Normal (a) and tangential (b) linear elastic-brittle law for broken interface [23].

to assume that it is compressed with the same stiffness as it had if it would not be broken ( $\sigma \leq 0$ ). However, under shear loads, the spring is structurally useless  $\sigma = \tau = 0$ .

$G_{Ic}$  and  $G_{IIc}$  are the energy release rate (ERR) in pure modes I and II, respectively. The ERR, also known as fracture energy, is defined in this model as the stored elastic strain energy per unit area in the unbroken spring at the crack tip. In a mixed mode crack, the ERR can be defined as Equation 2.5 shows.

$$G(x) = G_I(x) + G_{II}(x) = \frac{\langle \sigma(x) \rangle_+ \langle \delta_n(x) \rangle_+}{2} + \frac{\tau(x) \delta_t(x)}{2} \quad (2.5)$$

The operator  $\langle \cdot \rangle_{\pm}$  is referred as Macaulay brackets or ramp function and it is defined as  $\langle \delta \rangle_{\pm} = \frac{1}{2}(\delta \pm |\delta|)$ .

This way, springs that are undamaged store elastic strain energy under tangential or tension loads, but not compression because compression loads reduce the possibility to create or propagate cracks, which are provoked due to tension and shear. Then,  $G_I(x) = 0$  for  $\sigma(x) \leq 0$ .

On a separate issue, the critical normal and shear stresses ( $\sigma_c$ ,  $\tau_c$ ) depend on the fracture mode mixity, which is characterized by different fracture mode angles: energy based angle  $\psi_G$ , stress angle  $\psi_{\sigma}$  and relative displacement angle  $\psi_u$ . Furthermore,  $\delta_{nc}(\psi(x))$  and  $\delta_{tc}(\psi(x))$  are the critical normal and tangential displacements.

It must be noticed that all the critical variables ( $\sigma_c(\psi(x))$ ,  $\delta_{nc}(\psi(x))$ ,  $\tau_c(\psi(x))$  and  $\delta_{tc}(\psi(x))$ ) are functions of the fracture mode mixicity angle ( $\psi$ ) at a particular point  $x$ . As it is defined in [20], the different fracture mode mixicity angles are related between each other (see Equation 2.6). In fact, they coincide in pure fracture modes I and II.

$$\tan(\psi_G) = \sqrt{\frac{k_n}{k_t}} \tan(\psi_{\sigma}) = \sqrt{\frac{k_t}{k_n}} \tan(\psi_u) \quad (2.6)$$

Finally, if  $\tan^2(\psi_G) = \frac{G_{II}(x)}{G_I(x)}$  for  $\sigma(x) > 0$ , we can write Equation 2.5 as follows:

$$G(x) = G_I + G_{II} = G_I(1 + \tan^2(\psi_G)) \quad (2.7)$$

Therefore, each spring at a point  $x$  might have different critical values and a crack will propagate whenever the fracture energy reaches its critical value, which is defined as  $G_c = G_{Ic} + G_{IIc}$ .

Finally, in this project, we will follow the phenomenological law for the interface fracture energy as function of mode mixicity proposed in [5] by Hutchinson and Suo, just like it is done in [20], [23] and [22]:

$$G_c = G_{Ic}[1 + \tan^2((1 - \lambda)\psi_G)], \quad (2.8)$$

where  $G_{Ic}$  is the fracture energy in pure opening mode I and  $\lambda$  is a fracture mode-sensitivity parameter that characterizes mixed mode interfaces. Typical values for this parameter are comprised in [0.2,0.3] interval. For example,  $\lambda = 0$  means that crack advance depends only on the mode I component of ERR.

Furthermore, the LEBIM is implemented by means of a User Subroutine of material (UMAT), which is formulated in plane strain in our case. A subroutine permits to model any kind of behaviour law in FORTRAN language. For every iteration, ABAQUS provides the stresses, strains and predicted strain increments to the subroutine. Meanwhile, the UMAT returns the jacobian matrix and the stresses updated. By means of the state variable, the UMAT verifies whether an element is damaged or not, taking it into account to calculate the stiffness of the interface and, later, the jacobian.

For deeper understanding of the UMAT subroutine formulation and implementation, [9] should be consulted.

## 2.4 Finite fracture mechanics (FFM)

One hypothesis from Griffith criterion in classical fracture mechanics supposes infinitesimal crack growths. However, it was observed in many experiments that finite crack growths were usual. For this reason, Finite Fracture Mechanics (FFM) appeared based on the elimination of this hypothesis ([4]). Many energetic criteria were proposed through incremental energetic balances to justify failures under singularities, whereas stress criteria were used for failure predictions without defects. Some of the proposed criteria were: point-wise stress criterion, averaged stress criterion or incremental energy criterion.

### 2.4.1 Coupled criterion of finite fracture mechanics (CCFFM) by means of LEBIM formulation

There are evidences in the literature which ensure that stress and energetic criteria are necessary but not sufficient conditions for cracks to appear. In fact, both need to be fulfilled simultaneously. Under this premise, Leguillon ([7]) proposed a couple energetic and stress criterion which was able to predict failures under axial forces. This new line of research was denominated as Couple Criterion of Finite Fracture Mechancis (CCFFM).

It must be pointed out that Griffith's classical formulation of LEFM was successful without the consideration of stresses because crack growths were only subjected to continuous infinitesimal advances and infinite stresses at the crack tip. For this reason, any stress criteria would have been fulfilled. However, to consider discontinuous advances such as material breakage away from crack tips, stress condition must be incorporated to guarantee crack onsets at high stress regions.

The energetic criterion (see Equation 2.9) was based on incremental energetic balances, in opposition to the differential balances from Griffith theory. The increment considers the states before and after the onset of a new crack surface ( $\Delta S_c$ ).

$$\Delta\Pi + \Delta K + \Delta R = 0, \quad (2.9)$$

where  $\Pi$  is the potential energy, defined as the sum of the stored strain energy ( $U$ ) and the potential energy of the external loads ( $W$ );  $K$  is the kinetic energy and  $\Delta R$  is the energy dissipated associated to the creation of a new crack surface. It must be said that  $\Delta W = 0$  for displacement controls. Therefore, assuming quasi-static process and neglecting heat exchange (as it is down in [11]), the incremental energy condition of Equation 2.10 can be obtained. This expression indicates that the potential energy released to create the new surface should be larger or equal than the dissipated energy so the crack propagates from initial state to the final one.

$$-\Delta\Pi(\Delta S_c(a)) \geq \Delta R(\Delta S_c(a)) \quad (2.10)$$

Another way to see this expression is by the definition of the total energy, which will be extensively used in this project, as the sum of  $\Pi$  and  $R$ . What is observed is that the sum of their variation should be null or negative at the crack onset to induce the crack growth ( $\Delta\Pi + \Delta R \leq 0$ ).

A way to obtain the decrease of potential energy and the energy dissipated is by means of the ERR and fracture toughness, respectively. This way, Equation 2.10 can be expressed as follows:

$$\int_0^{\Delta a} G(a) da \geq \int_0^{\Delta a} G_c(\psi(x)) dx, \quad (2.11)$$

where  $G(a)$  is the ERR associated to the crack tip ( $a$ ) in the interface, which is obtained by Equation 2.5; and  $G_c(\psi(x))$  is the fracture energy associated to the same point  $x = a$ , which depends on the fracture mode mixity and it is obtained by means of the Hutchinson and Suo law, as it has been previously mentioned.

On the other hand, going back to LEBIM formulation, the stress criterion can be expressed as follows:

$$\frac{t(x)}{t_c(\psi(x))} \geq 1, \quad \text{for all } x, \quad 0 \leq x \leq \Delta a, \quad (2.12)$$

where  $t(x) = \sqrt{\sigma^2(x) + \tau^2(x)}$  is the stress vector modulus at point  $x$  of the interface and  $t_c(\psi(x)) = \sqrt{\sigma_c^2(\psi(x)) + \tau_c^2(\psi(x))}$  is the critical stress vector modulus, which is obtained from the maximum allowed stress components that depends on the fracture mode mixity.

The stress condition is evaluated before the crack onset on the entire interface component of the problem to predict where the crack is expected to initiate the growth. The aim is to localize one or several region where the interface might be damaged, for example, from  $x = 0$  to  $\Delta a$ , as Equation 2.12 shows.

Finally, reformulating the problem by means of dimensionless parameters (see [14] for further understanding) we can simplify the (pointwise) stress criterion (like [7]) and energetic criterion in Equation 2.13 and Equation 2.14, respectively.

$$s(x) = \frac{t_c(\psi(x))}{t(x)} \quad (2.13)$$

$$g(\Delta a) = \frac{\int_0^{\Delta a} G_c(\psi(\xi)) d\xi}{\int_0^{\Delta a} G(a) da} \quad (2.14)$$

The dimensionless parameter  $\mu = \frac{2G_{c, ch} k_{ch}}{\sigma_{c, ch}^2}$  characterize the interface brittleness and helps to uncouple  $\sigma_c$  and  $G_{Ic}$  (see [12], [14] and Figure 2.5). It should be noted that we use  $\mu$ , which considers the interface fragility, in opposition to parameter  $\gamma$ , which can be found in the literature (like in [10]) that refers to the fragility of all the materials involved in the problem. The relationship between those parameters is  $\mu = 2\gamma^2$ , just like [12] shows. The subscript  $ch$  refers to the characteristic properties of the interface (for example,  $G_{c, ch} = G_{Ic}$  for mode I) and it is usually chosen according to the predominant fracture mode of the problem.

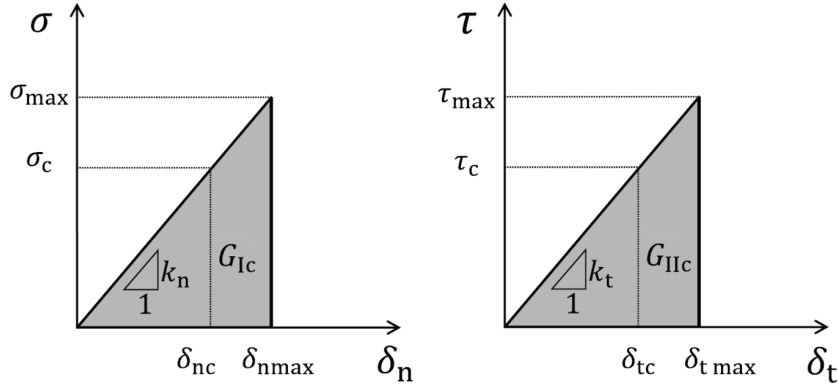
Figure 2.5 shows the constitutive law for CCFFM applied to LEBIM in mode I and mode II, just like it was done in [12]. They show that  $\mu$  can also be defined as  $\mu = \frac{\sigma_{max, ch}^2}{\sigma_{c, ch}^2}$ , where  $\sigma_{max, ch}^2$  and  $\sigma_{c, ch}^2$  are referred to the energy and stress criterion, respectively. This way, for the case  $\mu = 1$  and, therefore,  $\sigma_{max, ch}^2 = \sigma_c^2$ , the model is equivalent to the original LEBIM. In addition, it can be seen that when  $\mu$  increases the interface becomes stiffer.

Furthermore, this parameter helps us to uncouple the values of the interface parameters  $\sigma_{c, ch}$ ,  $G_{c, ch}$  and  $k_{ch}$ , taking into account that, in the original LEBIM, they were related (as can be deduced from Figure 2.3) by Equation 2.15 and Equation 2.16.

$$G_{Ic} = \frac{\sigma_c^2}{2k_n} \quad (2.15)$$

$$G_{IIc} = \frac{\tau_c^2}{2k_t} \quad (2.16)$$

Last, Equation 2.13 and Equation 2.14 can be combined into a single equation (Equation 2.17).



**Figure 2.5** CCFFM+LEBIM constitutive law in pure fracture (left) mode I and (right) mode II [12].

$$\frac{\sigma_{\text{nom},c}}{\sigma_{c,\text{ch}}} = \min_{\Delta a} \max \{s(\Delta a), \sqrt{\mu g(\Delta a)}\}, \quad (2.17)$$

where  $\sigma_{\text{nom}}$  is the minimum load that satisfies both criteria and produces a crack of length  $a_{\text{crit}} = \Delta a$ .

#### 2.4.2 Principle of minimum total energy subjected to stress condition (PMTE-SC)

Just like Mantic exposes in [11], the present formulation of the Principle of Minimum Total Energy subjected to Stress Condition (PMTE-SC) can be understood as a generalization of LEFM relaxing the too restrictive condition of  $\sigma_c \rightarrow \infty$  in order to be able to assess crack onset at weak singularities, stress concentrations, etc.

Many ways to obtain the variation of potential energy (or released energy) have been employed in the last years. However, the most general procedure ([16] and [15]) is described and used in this project, consisting on computing this value by means of Finite Element Method (FEM) codes at both states and subtracting them to later calculate the minimum (Equation 2.18) which indicates the crack behaviour.

$$\min_{\Delta a \in A_\sigma} \Delta \Pi(\Delta a) + \Delta R(\Delta a) \quad (2.18)$$

This expression intends to consider every crack possibility that belongs to the region where stress criterion is satisfied and the situation with minimum total energy is the one that occurs.

The issues of this method are that fine meshes might be needed in order to obtain precise values so the variation is accurately obtained and many configurations should be *randomly* analysed to find the minimum. This last sentence is not absolutely true in the sense that theoretically, the energy of many configurations should be obtained to reach the minimum. However, many systems behaves monotonously, so the minimum converges and it can be smartly obtained without random approaches.

The fact that the minimum is typically searched in small function regions (where stress criterion is fulfilled) leads to finite crack advances whose jumps may be associated to a tunneling effect through a total energy barrier ( $\Delta \Pi + \Delta R = 0$ ) by breaking the material bonds across a finite area subjected to sufficiently high stresses prior to fracture. This concept is easily understood by means of the next figure. Figure 2.6 shows total energy with respect to crack length of some typical situations that can be obtained when the PMTE-SC is applied. The red line is the region where the stress criterion is satisfied and it is represented as  $A_\sigma$ . The initial crack length and potential starting point of the crack is represented as  $a_0$ . In (a), the total energy at final stage is higher than in initial state so the crack does not grow although stress criterion is fulfilled. In (b), the stresses of the potential crack surface are high enough to allow tunneling through the total energy barrier, but the region is small and there is no excess of energy to create a larger crack. In (c), stresses are high in a large region so an unstable crack growth appear until the minimum

total energy. Meanwhile, in (d), stresses are high in small region with descending total energy so there is an unstable crack growth in the whole region.

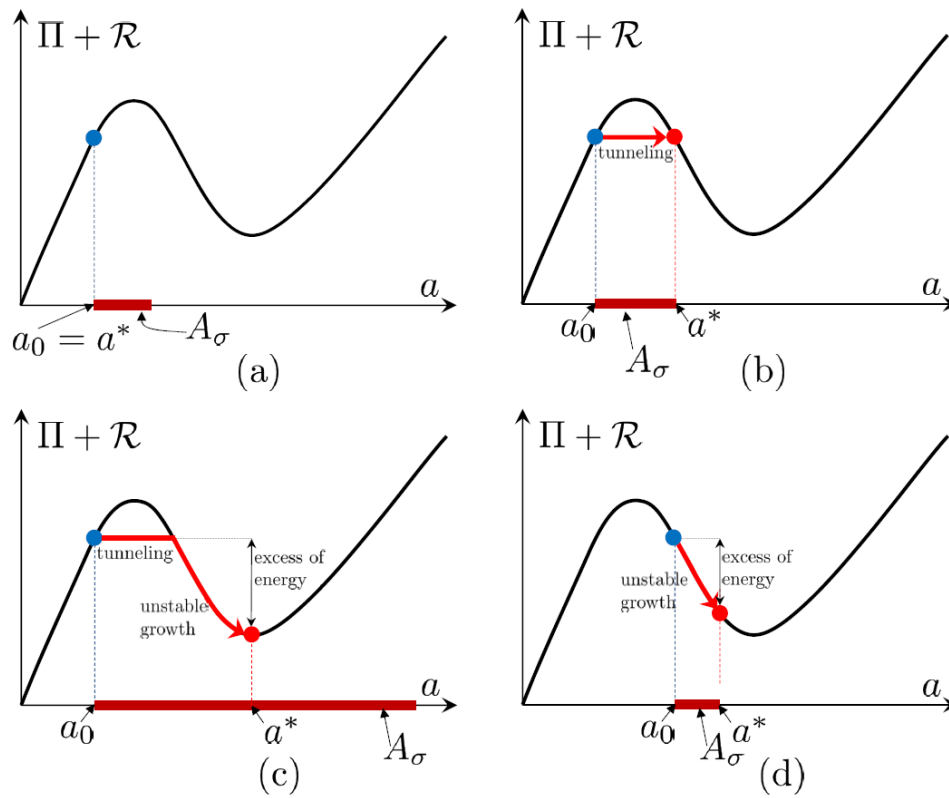


Figure 2.6 Examples of application of PMTE-SC. Total energy vs crack length [11].

# 3 Numerical implementation

*Eighty percent of success is showing up.*

WOODY ALLEN

In this section, the numerical implementation of the just mentioned methods will be explained, although further details can be looked over [15] and following sections of this project.

The formulation of CCFFM in which this project is based was introduced by [11] and numerically implemented for the first time in [15] to predict onset and propagation of cracks on Winkler's (spring) interfaces under a Double Cantilever Beam (DCB) test. The codes are written in Python language and they are based on FEM solution of ABAQUS. The interface is characterized by LEBIM and inserted by means of the user material subroutine UMAT.

Some concepts and their numerical implementation will be made below. The nomenclature and example (DCB with displacement or load control) employed in [15] will be used (see Figure 3.1). The DCB consists of a joint with two adherents and an adhesive layer between them, although further details are explained in chapter 4.

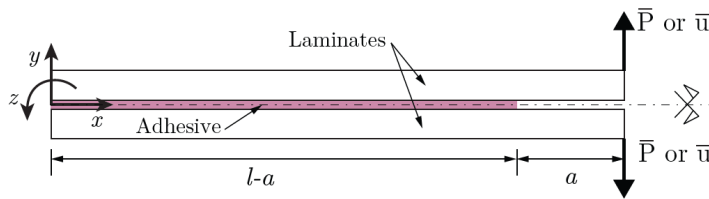
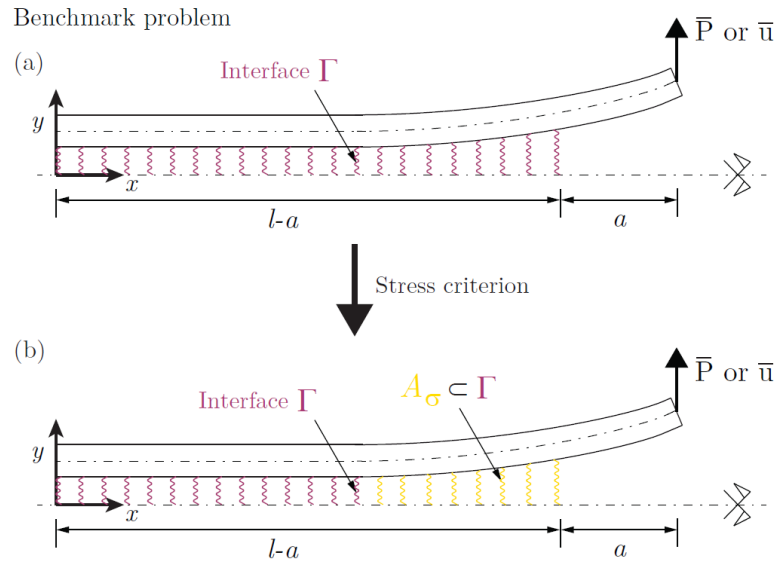


Figure 3.1 Benchmark problem: Double Cantilever Beam test [15].

## 3.1 Nomenclature and minimization subsets

Now, some explanations around the nomenclature employed will be made. The current crack length is referred as  $a$ .  $\Delta a$  is understood as a subset of the interface that might be composed of one or several crack advances and represent the points that are damaged in these crack advances. In addition,  $\Delta a_{crit}$  is the minimization solution of the energy criterion. The idea of the PMTE-SC is that new crack advances can only appear in regions where sufficiently high stresses are reached before fracture, being the stress criterion a filter to easily (and accurately) obtain the minimum total energy. This 'accurately' term is employed because the energy criterion could predict wrong crack propagation for low stress regions. Therefore, the stress criterion is the first one to be checked in the numerical implementation.

Just like Figure 3.2 shows, the interface is referred as  $\Gamma$ , while the set of undamaged points that satisfy the stress criterion and are candidates to initiate (or continue) the crack growth are referred as  $A_\sigma \subset \Gamma$ , which could be discontinuous.



**Figure 3.2** Interface casuistry. (a) Undamaged points of interface  $\Gamma$ ; (b) points satisfying the stress criterion ( $A_\sigma \subset \Gamma$ ) [15].

Lets also introduce  $\alpha(x)$  as the damage variable for each point  $x$  of the interface. This variable value is 1 for a fully damaged point and 0 for an undamaged one.

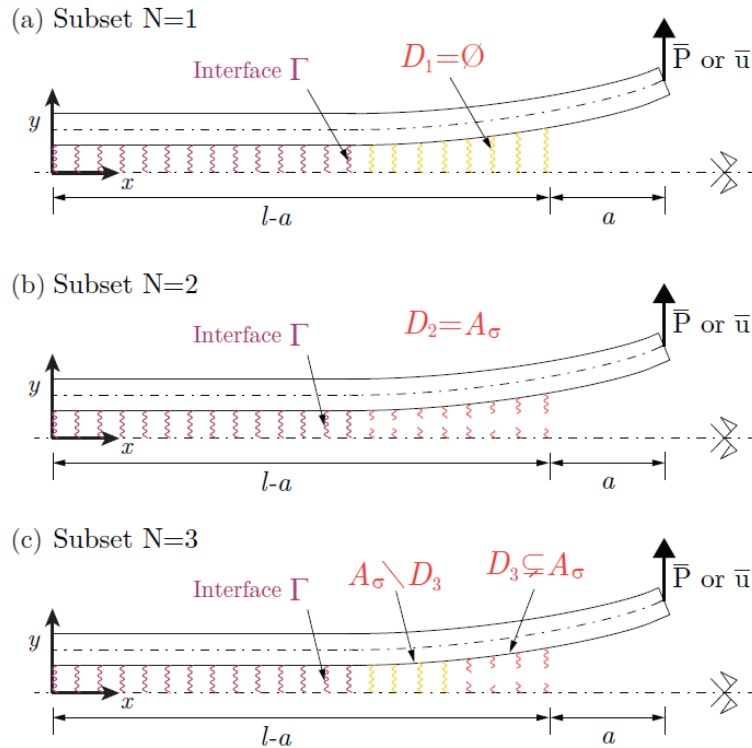
In this project, the minimization process employed is called Alternating Minimization Algorithm (AMA), which is a time-stepping procedure that exploits the separate convexity of the Total Energy with respect to displacements and damage variable ( $\alpha$ ). Thus, local minima are easily obtained and the challenge is to find the global one from them. To achieve it, it is crucial to explore different damaged areas within  $A_\sigma$  and to conveniently choose them. In Figure 3.3, three subsets are considered: the empty set, the full set and an intermediate case, respectively.

Each set considers a certain number of damaged points from all the candidates contained in  $A_\sigma$ . The set can be represented as  $D_n = \{x \in A_\sigma \mid \alpha_n(x) = 1\}$ , taking into account that any  $D_n$  is a subset of  $A_\sigma$ . The chosen sets are used to initiate the minimization process. Every subset where damage could initiate should be considered. The minimization solution might (or might not) coincide depending on the initialization set. For example, for a completely convex Total Energy curve, no matter the initialization subset, the minimum will coincide in the minimum point of the curve for any start; but for a concave curve, the minimums will be at the ends and they will differ. In fact, for complex problem, where Total Energy are expected to have several peaks, the desired solution might require a higher number of subsets. The global minima will be the minimum of the different solutions and the crack length will be the one associated to that energy.

It is clearly seen the importance of choosing the sets properly. As it is expected, the null and full sets should be at least considered. In fact, in this project, they are the two only sets that will be taken into consideration. However, the code is already prepared to take into account higher number of initial sets ( $N > 2$ ) so future studies might (and should) increase this number, although computational time will considerably increase.

In Figure 3.4, the already mentioned nomenclature and some additional terms used in the following explanations are compiled.





**Figure 3.3** Possible configurations of initial subsets [15]. (a) empty set; (b) full set; (c) intermediate case with damaged and undamaged points.

## 3.2 Iterations

The implementation is performed by means of several indexes (non-negative integers) that iterate around different aspects, mainly loads and minimization process. They are  $k$ ,  $m$ ,  $n$  and  $j$ . Each index runs for all its range before the previous indexes are increased. For example, for  $k=1, m=1, n=1$ ; every  $j$  is run before  $n$  increases to 2; when  $j$  and  $n$  are finished,  $m$  increases to 2 (if it applies) and finally, the next load step can be done ( $k=2$ ). They are explained below:

- **Index  $k$ :** Load step.  
This index refers to the load step.  
The load, which refers to load or displacement, is imposed by the user by means of different steps, while the algorithm provides the produced damage ( $\alpha$ ) and displacement field under that load. In particular, the load is introduced by means of an increment ( $\Delta f$ ) that is added to the previous load. The first load (which corresponds to  $k=1$ ) is obtained as the required one to initiate damage at one point of the interface according to infinitesimal LEBIM with  $\mu = 1$ .
- **Index  $m$ :** iteration within a load step for consecutive damage  
For each load step, several consequent crack advances can appear. For this reason, the  $m$  index is introduced, which makes another calculation for the same load but for the new configuration with larger damage. For example, imagine in the second load step ( $k=2$ ) where there is no initial damage, under a certain load, 30% of the adhesive breaks. Then, another  $m$  step is analysed ( $m=2$ ) with initial damaged configuration but under same load.  
The algorithm exits from the  $m$  loop when there is not new damage. In that moment, next load step is started. However, the maximum number of  $m$  steps can be defined by the user. For displacement control, only one iteration is done according to  $m$  index because a displacement can cause a certain damage but it will not increase for that fixed displacement. On the contrary, in load control, for a

Variables used in the pseudocode:	
$f$	Imposed load or displacement in the study case.
$\Delta f$	Load increment between two steps.
$\alpha(\text{element})$	Damage variable for each element along the interface. 1 value means damaged interface element (with zero stiffness in tension and shear). 0 value means undamaged interface element (with full initial stiffness).
$\alpha$	Damage configuration of $\Gamma$ given by all values $\alpha(\text{element})$ .
$u$	Displacement field
$\psi$	Fracture mode mixity.
$t_c(\psi)$	Critical traction modulus for the stress based criterion.
$G_c(\psi)$	Critical fracture energy for the energy based criterion.
$A_\sigma$	Set including all undamaged elements that satisfy the stress criterion
$D$	Proposed subset of $A_\sigma$ that contains damaged elements assumed before the minimization. Note that $A_\sigma \setminus D$ is composed for the rest of the undamaged elements of $A_\sigma$ .
$N$	Number of considered starting damage configurations.
$\epsilon$	Difference of the damage variable between two AMA iterations.
Each variable defined above may be affected by the following subscripts and superscripts:	
$k$	Load step.
$m$	Iteration within a load step.
$n$	First damage configuration within an iteration. $n \in [1, N(k, m)]$ , with $N(k, m) \geq 2$
$j$	AMA iteration

**Figure 3.4** Nomenclature used in pseudocode [15].

certain force, the damage can propagate permanently, reaching an unstable behaviour. If a snap-through behaviour is desired to be captured, the maximum number of  $m$  steps must be set to 1 and next load is obtained as in the first load step but for the current damaged configuration, which means it will be lower.

- Index  $n$ : first damage configuration within an iteration.  
For each  $m$ , the Total Energy minimization is performed by means of  $n$  and  $j$  indexes. As previously said, in this project,  $N=2$  so two  $n$  steps are considered: the empty set and the full set.
- Index  $j$ : AMA iteration  
Each step of the minimization process of a certain configuration ( $D_1$  and  $D_2$ ) is referred as  $j$ . The minimization is performed until the damage variable (or crack length) between two consecutive  $j$  steps coincide. The damage for the case  $j=1$  corresponds to the damage set in  $D_n^{k,m}$ .

### 3.3 Algorithm process

The general pseudo-code for an Alternating Minimization Algorithm for PMTE-SC will be briefly explained below. For a better understanding, a diagram from [15] is represented in Figure 3.5.

- In the first part of the code, the interface zone that satisfies the stress criterion  $A_{\sigma}^{k,m}$  under the load of step  $k$  is obtained. From this set, the two subsets ( $D_1^{k,m} = 0$  and  $D_2^{k,m} = A_{\sigma}^{k,m}$ ) are considered.
- In the second part, the minimization of the total energy is carried out for each starting subset  $D_n^{k,m}$  with their initial damage configuration  $\alpha_n^{k,m}$ . The AMA splits the process into two separated minimizations: one with respect to the displacement field  $u_n^{k,m}$  and another with respect to the damage variable  $\alpha_n^{k,m}$ . Therefore, each  $j$ -th iteration is divided into two:
  - First, the displacement field  $u_{n,j}^{k,m}$  is obtained by minimizing the total energy for the initial damage configuration defined by  $D_n^{k,m}$ . It must be mentioned that the dissipated energy does not change in this minimization because it does not depend on the displacement field. Therefore, the FEM provides the displacement field  $u_{n,j}^{k,m}$  that produces the minimum potential energy  $\Pi(f^k, u_{n,j}^{k,m}, \alpha_{n,j-1}^{k,m})$  for the damage obtained in previous  $j$  step ( $j-1$ ). For  $j=1$ , the damage corresponds to the set  $D_n^{k,m}$ .
  - Secondly, for the just obtained displacement field, the total energy is minimized with respect to the damage configuration  $\alpha_n^{k,m}$ . This is done by a loop over every element in  $A_{\sigma}^{k,m}$  comparing the change in elastic strain energy and dissipated energy. This method can be applied because the expression to be minimized is linear with respect to the damage.

The AMA is repeated until no difference between consecutive steps is achieved for all  $D_n^{k,m}$ . Then, solutions for all  $D_n^{k,m}$  are compared and the configuration with  $u_n^{k,m}$  and  $\alpha_n^{k,m}$  that provides lowest total energy ( $\Pi + \Delta R$ ) is chosen. Note that it is the potential energy (and not the variation) the value to be minimize because  $\Pi^{k-1}$  does not vary under a certain load.

This process continues around  $m$  iterations until no more damage is propagated for a certain load, which is related to step  $k$ .

## 3.4 DCB example

Herein, we will get back to the DCB example from [15] to better understand the internal process of the algorithm.

### 3.4.1 Displacement control

Figure 3.6 shows, for displacement control, the total energy graphs which compares the points from the numerical tool and the curves, which are the expected analytical solution.

Some comments are made:

- The discontinuous part of the curves represent the interface zone that fulfills the stress criterion, whereas the continuous part corresponds to the interface part that does not and, therefore, is not candidate to brake.
- The first step (first load) is represented in green and the second one in red.
- To be able to identify the output points, the indexes  $kmnj$  are represented near them.
- Filled circles represent points with  $n=1$ , whereas empty circles corresponds to points with  $n=2$ . Finally, half-filled circles represent both cases, if they coincide.

We can perceive several aspects from the figure:

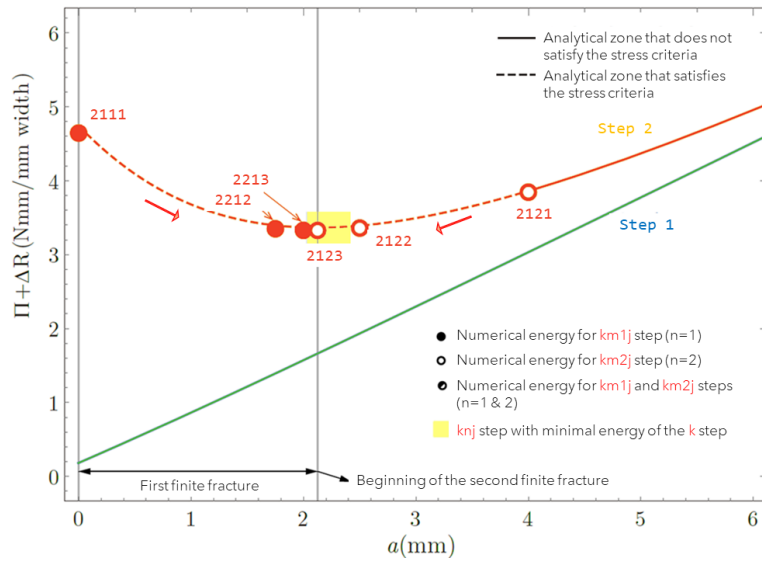
- The left-hand point (2111), represents  $k=2$ ,  $m=1$ ,  $n=1$ ,  $j=1$ , which is the initial point of the minimization for the empty set, whereas the right-hand point (2121) is for the full set.
- Next  $j$ -th points are 2112 and 2122, respectively, which are intermediate points of each minimization process.

```

for  $k$  step do
  The iteration  $m$  is initialized with  $m = 1$  ;
  if  $k=1$  then
    The variables are set:  $\begin{cases} f^k \text{ selected based on infinitesimal LEBIM with } \mu=1 \\ \alpha^k = \emptyset \text{ and } \alpha^{k,m} = \emptyset \\ u^k = \emptyset \text{ and } u^{k,m} = \emptyset \end{cases}$ 
  else
    The variables are set:  $\begin{cases} f^k = f^{k-1} + \Delta f \\ \alpha^k = \emptyset \text{ and } \alpha^{k,m} = \alpha^{k-1} \\ u^k = \emptyset \text{ and } u^{k,m} = \emptyset \end{cases}$ 
  while  $\alpha^k = \emptyset$  do
     $m=m+1$  ;
    Minimize  $u^{k,m-1} \mapsto \Pi(f^k, u^{k,m-1}, \alpha^{k,m-1})$  //FEM calculation//;
    Update  $u^{k,m} = u^{k,m-1}$ ;
    From  $u^{k,m}$  compute  $\psi^{k,m}$  and then:  $t_c(\psi^{k,m})$  and  $G_c(\psi^{k,m})$  on  $\Gamma$  ;
    Set  $A_\sigma^{k,m}$  on  $\Gamma$  by stress criterion ;
    if  $A_\sigma^{k,m} = \emptyset$  then
      | Set solution  $u^{k,m}$  and  $\alpha^{k,m} = \alpha^{k,m-1}$ 
    else
      Define  $D_n^{k,m} \subset A_\sigma^{k,m}$  for  $n \in [1, N(k, m)]$ , with  $N(k, m) \geq 2$ ;
      Define the first  $\alpha_n^{k,m}$  for every  $N$ ;
      for  $n = 1$  to  $N(k, m)$  do
         $j = 1$ ;
         $\alpha_{n,j}^{k,m} = \alpha_n^{k,m}$ ;
        while  $\|\alpha_{n,j}^{k,m} - \alpha_{n,j-1}^{k,m}\| \leq \epsilon$  do
           $j = j + 1$ ;
          minimize  $u_{n,j}^{k,m} \mapsto \Pi(f^k, u_{n,j}^{k,m}, \alpha_{n,j-1}^{k,m}) + \Delta R(G_c(\psi^{k,m}), \alpha_{n,j-1}^{k,m} - \alpha^{k,m-1})$  //FEM calculation// ;
          minimize  $\alpha_{n,j}^{k,m} \mapsto \Pi(f^k, u_{n,j}^{k,m}, \alpha_{n,j}^{k,m}) + \Delta R(G_c(\psi^{k,m}), \alpha_{n,j}^{k,m} - \alpha^{k,m-1})$ 
          subject to  $0 \leq \alpha_{n,j}^{k,m}(\text{elem}) \leq \alpha^{k,m-1}(\text{elem}) \forall \text{ elements} \in A_\sigma^{k,m}$ 
          and  $\alpha_{n,j}^{k,m}(\text{elem}) = \alpha^{k,m-1}(\text{elem}) \forall \text{ elements} \notin A_\sigma^{k,m}$ ;
        Set solution  $u_n^{k,m} = u_{n,j}^{k,m}$  and  $\alpha_n^{k,m} = \alpha_{n,j}^{k,m}$ ;
      Store configuration  $[u_n^{k,m}, \alpha_n^{k,m}, \Pi(f^k, u_n^{k,m}, \alpha_n^{k,m}) + \Delta R(G_c(\psi^{k,m}); \alpha_n^{k,m} - \alpha^{k,m-1})]$ ;
    Set solution  $u^{k,m} = u_n^{k,m}$  and  $\alpha^{k,m} = \alpha_n^{k,m}$  for the  $n$  with the lowest value of
       $\Pi(t^k, u_n^{k,m}, \alpha_n^{k,m}) + \Delta R(u^{k,m-1}; \alpha_n^{k,m} - \alpha^{k,m-1})$  ;
    Set solution  $u^k = u^{k,m}$  and  $\alpha^k = \alpha^{k,m}$ ;

```

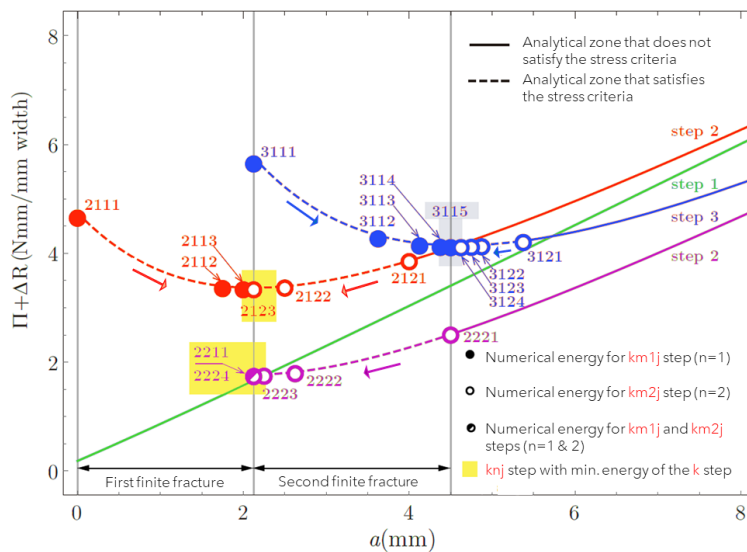
**Figure 3.5** General pseudo-code for an Alternating Minimization Algorithm for PMTE-SC [15].



**Figure 3.6** Analytical and numerical results for DCB example from [15] under displacement control. Total energy vs crack length. (Steps 1 and 2).

- Finally, the minimum total energies are obtained for  $j=3$  in 2113 and 2123. Although the analytical convex curve tells us it should converge to a unique point, it does not happen numerically due to the element size. In this case, the minimum falls within an element and each approach suggests different solutions. The one provided from  $n=2$  is the absolute minimum and, therefore, the chosen one. The crack length associated to this total energy is the crack growth that the adhesive (interface) will suffer.
- It can be concluded that the error we induce due to the numerical element discretization is up to the element size.

In Figure 3.7, the remaining part of step 2 (this is  $m>1$ ) and the following step 3 are represented.



**Figure 3.7** Analytical and numerical results for DCB example from [15] under displacement control. Total energy vs crack length. (Steps 1, 2 and 3).

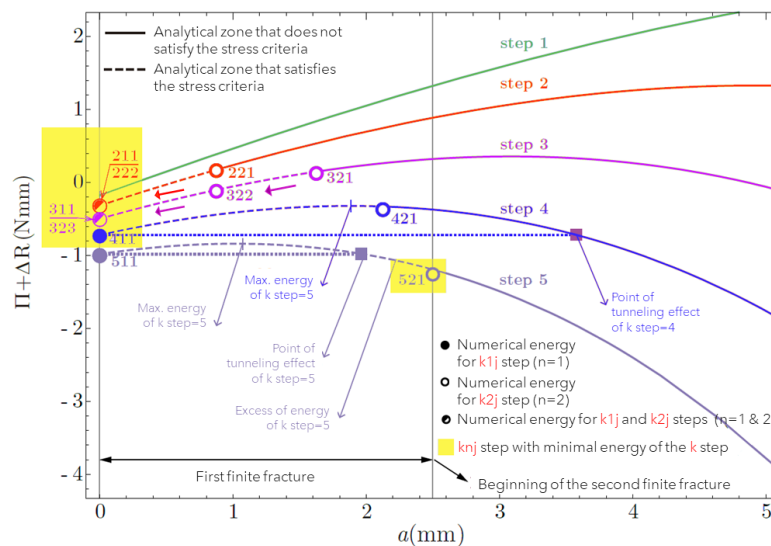
We can appreciate how, after the first finite crack another iteration within the  $k=2$  loop is done for  $m=2$ , which refers to the possibility that for the same load, a second crack appears. In this case, it is seen that both minimizations ( $n=1$  and  $n=2$ ) reach the solution of the empty set, which means that the crack does not propagate under that load. It must be noticed that the pink curve (and points) corresponding to  $k=2$ ,  $m=2$  have lower energy than those from  $k=2$ ,  $m=1$  (red curve), this is because  $m=2$  already has damage and the adhesive has less capacity to store energy. In addition, the curve is monotonously ascending so crack will never appear for this displacement.

Then, the  $k=3$  iteration is performed and similar comments than the ones done above for  $k=2$ ,  $m=1$  can be applied, leading to a second finite crack. In this case, the AMA took  $5j$  iterations to reach the minimum for  $n=1$  and 4 iterations for  $n=2$ .

From these figures and comments we can see the importance of the element size and the load increment. In chapter 4, we will refer to these aspects and quantify their influence. However, it is already seen that to be able to capture an infinitesimal growth, the element size should be as small as possible.

### 3.4.2 Load control

Under load control, different behaviour is perceived because total energy curves are concave. They are represented in Figure 3.8 for steps 1 to 5 and Figure 3.9 for steps 1 to 13. Since  $m$  is limited to 1 to be able to capture the snap-through, the second index is removed and the figures only shows  $knj$  since  $m=1$  for every point.

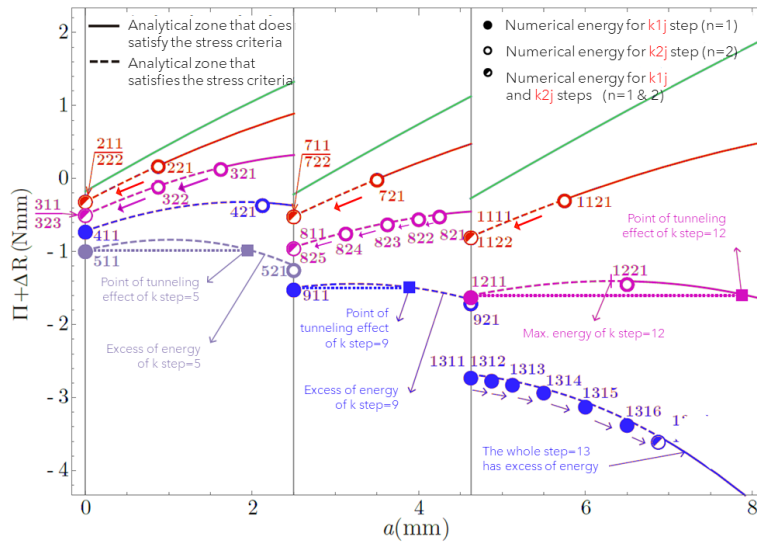


**Figure 3.8** Analytical and numerical results for DCB example from [15] under load control. Total energy vs crack length. (Steps 1 to 5).

In Figure 3.8, it is seen for  $(k)$ -th steps 2 to 4 that, although some elements fulfil the stress criteria, the minimum energy corresponds to an unbroken interface. However, in the case of step  $k=4$  (blue curve), the maximum of the curve is within the elements complying the stress criterion. Therefore, the minimization with  $n=2$  leads to the right-hand point (421), which is a local but not an absolute minimum. For a higher load, in step  $k=5$  (lilac curve), the situation is the opposite, it is the minimization with  $n=1$  the one that ends in the left-hand point (511) but the energy is higher than the minimum of  $n=2$  (521). Therefore, for this load there is a crack advance. The excess of energy means that if the energy of points 511 and 521 coincide so they are contained in the horizontal dashed line, the crack would have also appeared and this excess would have induced a larger crack up to the total number of elements fulfilling the stress criterion.

In Figure 3.9, the following load steps of the problem are shown. Another crack advance takes place at step  $k=9$  (central blue curve) and, finally, at step  $k=13$  (right-hand blue curve). Whereas steps 6 and 10

(green lines) correspond to the restarted loads that help to capture the snap-through instability. Elements does not fulfil the stress criterion under this load.



**Figure 3.9** Analytical and numerical results for DCB example from [15] under load control. Total energy vs crack length. (Steps 1 to 13).

It is interesting to point out how the curves are concave but their behaviour is different depending on the load value.

- For small loads, the curve is monotonously ascending so cracks will never appear, no matter the stress criterion.
- For intermediate loads, the curves have ascending and descending parts so depending on the stress criterion and energy values, cracks might (or might not) appear.
- Finally, for high loads, the curve is monotonously descending (like in step 13) so the crack advance will be equal to the total length of adhesive that fulfils the stress criterion.

For further understanding of total energy behaviour, section 4.6 for DCB and subsection 5.3.3 for DLJ could be consulted.





## 4 Tests: Double Cantilever Beam (DCB)

---

*The most important thing is to keep the most important thing the most important thing*

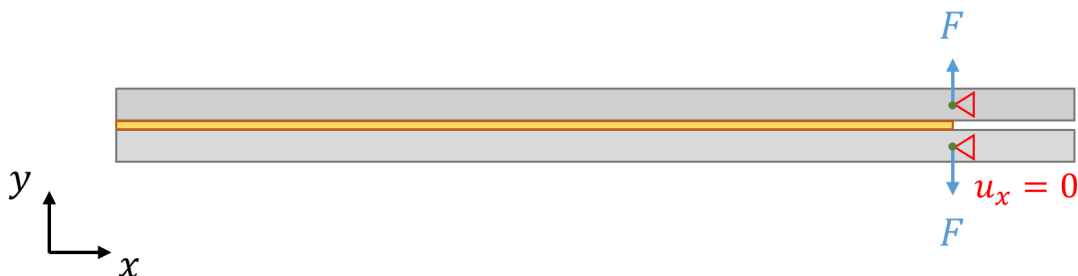
DONALD P. CODUTO

Once the numerical tool and main concepts are clear, two of the most common tests will be analysed in this project: the double cantilever beam (DCB) and double lap joint (DLJ).

The double cantilever beam (DCB) test is the most commonly used method to measure the initiation and propagation values of Mode I fracture energy ( $G_I$ ) under static and cyclic loads because its fracture is caused by pure mode I. Furthermore, it is one of the easiest joints to be tested, not only in the laboratory but numerically.

### 4.1 Problem description

The test consists on a tensile load, normal to the crack surface, which is applied to the specimen (two beams joined by an adhesive) at its free end. Figure 4.1 shows the sample that will be implemented in ABAQUS, just as the boundary conditions and loads.

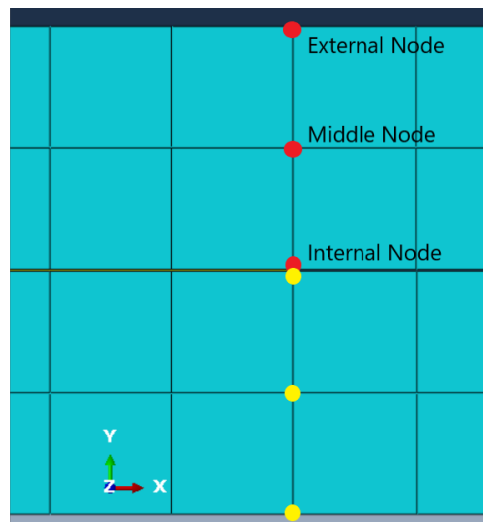


**Figure 4.1** DCB sample.

Due to this general simplicity, it will be the first joint to be considered in order to understand the existing tool and verify that the results coincide with what is theoretically expected. Some parameters like the step increment or the mesh size will be study. Several conclusions will be addressed around those parameters and the code will be further developed to increase accuracy.

## 4.2 Effect of the applied load location

Firstly, we will verify whether the node where the load (or displacement) is applied affects substantially to the results. In particular, we will make the analysis under displacement control and verify the reaction at the nodes. For the sake of simplicity, the element size that has been chosen is 0.75mm, which involves 2 elements in the beam thickness and, therefore, allows us to obtain the desired reaction in 3 different points (being one of them the centroid of the beam section). Figure 4.2 shows the nomenclature we will use for the nodes to be able to adequately refer to each analysis.



**Figure 4.2** Nomenclature for the beam nodes. In red for upper beam and yellow for lower beam.

This numerical analysis consist of two beams bonded to the adhesive with TIE interaction. The three parts (two beams and the adhesive) are modelled with two dimensional solid elements with plane strain formulation: CPE4.

The boundary conditions applied are:

- Restricted displacement in X direction for both reference nodes.
- Vertical displacement (Y direction) of 1mm for upper beam node and -1mm for lower beam node (Figure 4.3). Notice that, although the figure shows the BC for the middle node, this is equivalent for the analyses in which the displacement is imposed at the internal and external nodes by only changing the set where BC are applied.

Firstly, we must verify if the vertical displacement of the three nodes is the same. To do that, the analysis where the displacement is applied in the middle node is revised. It can be seen, in Figure 4.4, how the displacements of the external node (ID: 1) and internal node (ID: 2) are close to 1 but not the same. This clarifies that, even being the beam very thin compared to the longitudinal length, its rigidity affects. Although the variation might not seem too substantial, the reaction force should also be affect by the variation of the 3rd significant digit, which may involve tens of newtons.

By obtaining the reaction forces of the different analyses (Table 4.1), we confirm the variations are not negligible.

Trying to look for an alternative way to apply the load, we carried out an analysis where the displacement is imposed in a reference point (RP), located at the centroid of the beam. To transmit the condition, a kinematic coupling is applied. Although the reaction force obtained is closer to the desired (see last row of Table 4.1), it still involves an error of 0.22% with respect to the reference case, which is the load applied at the middle nodes. Therefore, it can be concluded that the analyses should be carried out, to the extent possible, with the load applied in a node at the centroid of it. In case a certain mesh without

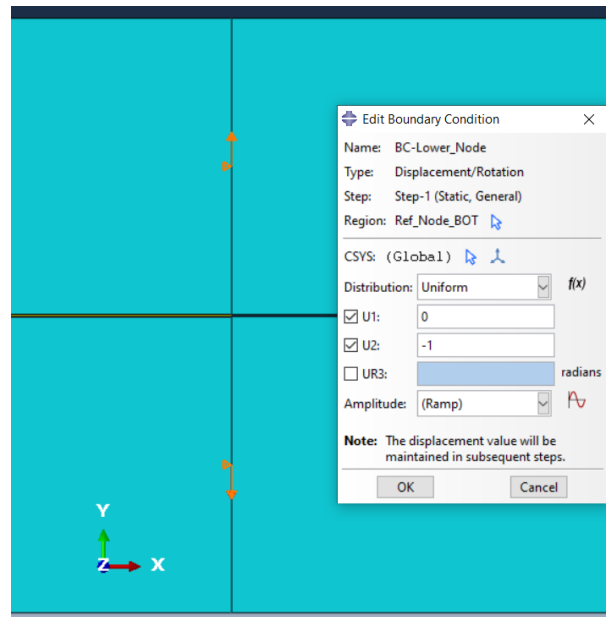


Figure 4.3 Boundary condition at reference nodes of the beams.

Part Instance	Node ID	Orig. Coords	Def. Coords	Attached elements	U, U2
VIGA_TOP	1	237, 1.51, 0	236.818, 2.506	1, 36	0.996625
VIGA_TOP	7	237, 0.76, 0	237, 1.76, 0	1, 2, 35, 36	1
VIGA_TOP	2	237, 0.01, 0	237.183, 1.005	2, 35	0.995186

Figure 4.4 Nodes' displacements for the analysis where the load is applied in the centroid.

Table 4.1 Reaction forces from analyses where displacement is applied in different points.

Displacement imposed at...	Reaction force [N]
External nodes	1176.99
Middle nodes	1185.18
Internal nodes	1180.37
RP in middle position	1187.76

such node is required, the model with the load applied at a reference point located at the centroid will bring more accurate results than loads applied at nodes which are far from the centroid. Any case, since we have the possibility to decide the mesh sizing, we will only choose those which include nodes in the centroids of the beams.

### 4.3 Parameter study: step increment

At this point, some studies will be carried out around the model.

Firstly, we have realized that the choice of the **step increment** has a substantial impact on the results. Therefore, a study around this parameter will be accomplished.

Both displacement and load controls have been analysed and represented in Figure 4.5. The curves of the figures correspond to different step increments. This way, for displacement control, we have  $\Delta U = 0.005, 0.010, 0.020\text{mm}$  (in blue, red and yellow lines, respectively), whereas for load control, the curves correspond to  $\Delta F = 0.5, 1, 2, 4\text{N}$  (in blue, red, yellow and purple lines, respectively). Every point of the

plot represents a value obtained from the numerical calculation and the spacing between them is equal to the step increment. On the other hand, the curves consist on linear interpolations between the just-mentioned points.

As it is expected, the first point different from (0,0), which belongs to the load required so 1 element satisfies the stress criterion (referred as *prediction of original LEBIM* in advance), coincides for every analysis; as well as the slopes of the curves, since it represents the stiffness of the specimen and it must be the equal if the mesh is the same. As it is appreciated, this stiffness is reduced after the appearance of damaged elements, which can be detected in the graphs as vertical lines in displacement control and horizontal lines in load control.

On the other hand, it is clearly observed that there is general disagreement in the points obtained between same models with the only difference of the step increment. This is because the code, briefly said for the sake of simplicity, verifies if there are damaged elements (they satisfy both stress and energy criteria) under a certain load (force or displacement) we introduce. Since this load is gradually increased at a constant increment that we 'randomly' introduce, the results will highly depend on that choice.

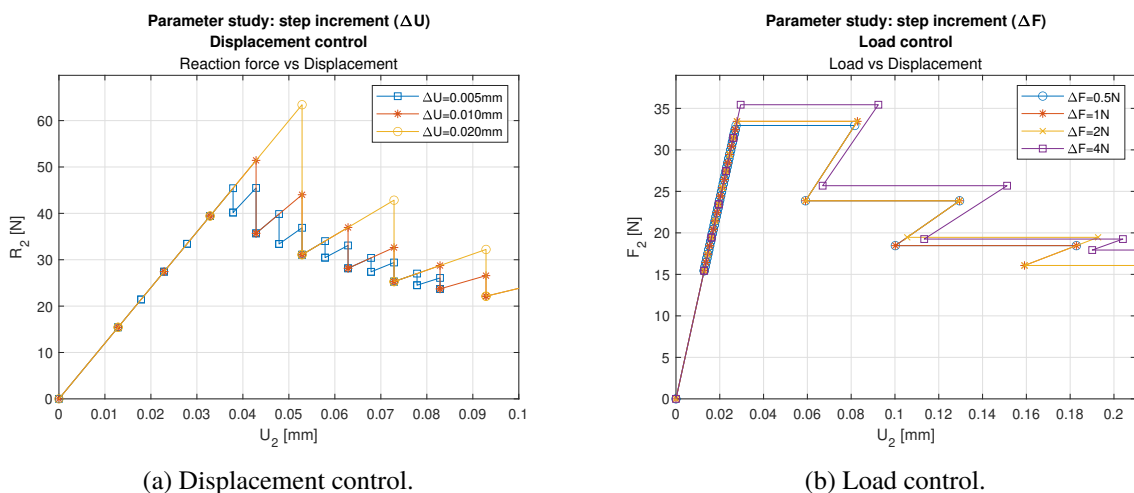


Figure 4.5 Parameter study: step increment.

For example, under displacement control (Figure 4.5a), the first failure appears when the ascending line drops. However, this point only means that at that displacement, there are damaged elements, but they might be damaged under a lower displacement and, in fact, they are. What is certainly known is that, for the previous load, there was no damage, so the minimum load that induces the first damage is contained in the interval between those two loads, although it is unknown. Therefore, depending on the chosen step increment, the interval where the failure occurs will be larger or smaller.

A similar behaviour appears under load control (Figure 4.5b), although in this case, the load increments are proportionally lower and it is more difficult to be appreciated.

For this reason, it seems reasonable to reduce this increment to increase accuracy. However, this measure involves, on the one hand, higher computational times and, on the other hand, the necessity to iterate for each problem since this increment will be appropriate for a certain problem but not for another one. Therefore, in section 4.5, the code will be developed in order to achieve a trade-off between more accurate results and low computational time and trying it to be valid for any problem without additional modifications.

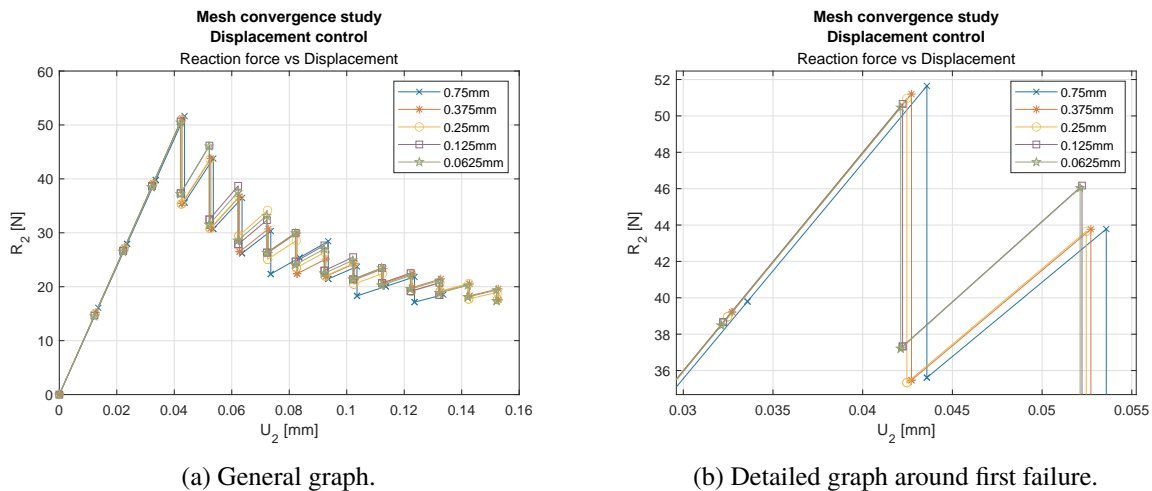
## 4.4 Mesh convergence study

In this section, a study to understand the mesh influence will be accomplished. In agreement with comments from section 4.2, mesh sizes involving even number of elements along the thickness of the beams will be chosen (see Table 4.2). This element size refers to the characteristic length of the elements, which is, for two dimensional elements of four nodes, the length of the square side.

**Table 4.2** Mesh convergence study. Size details.

Mesh No.	Element size [mm]	Elem. No in beam thickness
1	0.75	2
2	0.375	4
3	0.25	6
4	0.125	12
5	0.0625	24

Both displacement and load controls (Figure 4.6 and Figure 4.7, respectively) have been analysed. Just like before, every point of the plot represents a value obtained from the numerical calculation and the spacing between them is equal to the step increment. Since this increment is constant, the spacing between points for every curve is the same. On the other hand, the curves consist on linear interpolations between the just-mentioned points. Each figure represents a general view of the curve evolution and a zoomed view around the first failure so the differences between meshes are better perceived.



**Figure 4.6** Mesh convergence study. Displacement control.

We can see that it seems there is no substantial discrepancy between different mesh sizes with a fixed step increment ( $\Delta U = 0.01mm$  for displacement control and  $\Delta F = 1N$  for load control). This happens because for a certain load that induces a crack growth up to length  $\Delta$ , the number of elements involved will be different depending on the mesh, but the absolute length of the crack  $\Delta$  will be the same, taking into account the element sizes. For example, if for an element size of 1mm the predicted crack is  $\Delta = 3mm$ , which corresponds to 3 elements, and we reduce this element size to 0.25mm, the crack might have a length of 3, 3.25, 3.5 or 3.75mm, that is the precision allowed by the mesh size, but it will be smaller than 4mm, since for the previous analyses this length was not reached. Therefore, the only differences between analyses are due to number of elements representing the adhesive and numerical errors, such as small variations in the slopes or the calculation of the prediction of original LEBIM.

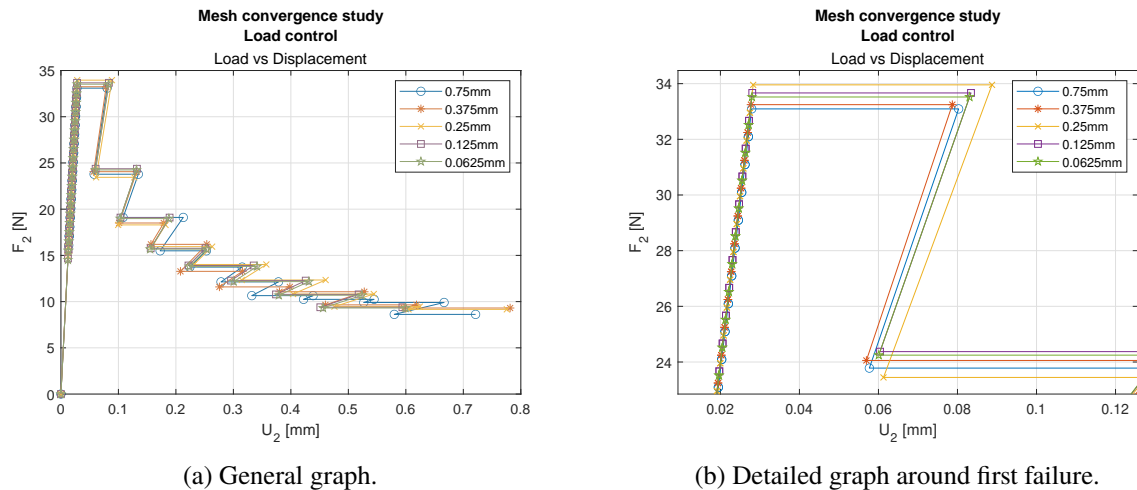


Figure 4.7 Mesh convergence study. Load control.

To conclude, it must be mentioned that the linear loading slope is the same for every step increment with same mesh sizes (Figure 4.5a and Figure 4.5b), since the adhesive and adherend properties have not changed and linear analyses imply that higher loads increase the displacement proportionally and, therefore, the slopes are constant.

However, for different mesh sizes (Figure 4.6 and Figure 4.7), there are some variations because the slope, which represents the structure stiffness, might slightly change, especially for coarser meshes. In particular, the biggest difference appears for the mesh size of 0.75mm, concluding that the chosen element size should be 0.375mm, which is equivalent to 4 elements in the beam thickness, or smaller. Anyways, from this study (and others that are not shown for the sake of brevity), we have seen that mesh has not a substantial effect on the results, although refined sizes provides more accurate measures of the crack length since the numerical error will be limited by the element size. In contrast, the failure prediction is highly influenced by the step increment since the applied load in each step depends on the prediction of original LEBIM and the increment. Therefore, more effort will be made around this parameter in order to increase accuracy. However, for other analyses or stiffness values, mesh could be more relevant.

## 4.5 Accuracy rising by code development

Taking into account that further studies will be based on obtaining an accurate value of the first failure load, we will spend some time and effort trying to improve the code around this aspect. There are two ways to proceed: reduce mesh size or reduce step increment. Both approaches would considerably increase the computational time required to solve the problem. However, we have already seen that decreasing the mesh size only helps to have an accurate crack length under a certain load, but it does not provide an accurate failure load value. On the other hand, decreasing the loading increment is the exactly approach needed to reach an accurate failure load value. Therefore, the code has been developed around this approach in order to improve the accuracy of the results and trying not to considerably increase the computational cost. Although several versions have been developed, only an intermediate and the definitive one will be mentioned.

Thanks to the conclusions obtained from Figure 4.5, it is already well known that the loading increment choice is crucial to predict the first failure. In case the problem which is being analysed is known and the order or magnitude too, the loading increment might be chosen in a sensible way. However, for many unknown problems, this will be randomly chosen and it will end up in inaccurate results or many iterations. In addition, the simple solution of reducing the loading increment up to very small values is either inappropriate since it involves high computational costs. In fact, since the problem is linear, small

increments are only required around the failure load, and not in the linear loading where only two points are needed to know the slope. Therefore, the objective is to arrive to a trade-off solution where the code iterates around the failure load until a certain accuracy is reached, but large increments are generally used far from this point.

The last commentary before explaining the new and innovative code developments should be reserved to remark that these thoughts and way of designing the adaptative model derive from the observation of Load vs Displacement graphs like the showed before and not from an energetic point of view, which will be addressed in section 4.6.

### 4.5.1 High accuracy model with fixed initial increment

This first approach is focused on obtaining the failure point accurately, although the number of steps were high, which will be addressed later on. This approach is based on loading step by step at a rate of the increment introduced by the user until reaching the failure load. After this, it recalculates this failure point by reducing the increment by half as many times as required to reach a certain tolerance. This tolerance is also chosen by the user. In next figures, a tolerance of 10% (TOL=0.1) will be used since it helps to clearly understand the process. The value that we compare with this tolerance is the quotient between the last increment and the initial one chosen by the user. This fast choice was made in order to focus on the main objective of achieving failure load with accuracy, but it still has the problem that depends on the initial 'random' choice. This will be addressed in THE next version.

The way of proceeding once the failure is reached and a more accurate value is desired is that the load is decreased by  $(incree/2)$  and two things can happen:

- It fails again. Then, if the *TOL* is not reached, it decreases again by  $(incree/2)$  up to fulfill the tolerance criterion; otherwise, it continues loading with the initial increment, just like the original code.
- It does not fail. Then, the suggested load is increased by  $(incree/2)$  and, in next step, it will be checked whether failure appears again (go to previous point) or not (increase by  $(incree/2)$  until failure).

This code improvement is summarized in Table 4.3 in the second column, where it can be compared with the original one in first column and the adaptative model, which will be explain in next section. In this table, LC and DC means load control and displacement control, respectively.

**Table 4.3** Comparison between different *PMTE-SC* code versions.

	Versions			
	Original	High accuracy	High accuracy and adaptative	
	LC & DC	LC & DC	Load control	Displ. control
Loading increment	Cst	Cst	$incree_k = Factor_{incree} \cdot incree_{k-1}$	
Incre proposal to reach acc. failure	-	$incree_k = 0.5 \cdot incree_{k-1}$	$incree_k = 0.5 \cdot incree_{k-1}$	
Tolerance criterion	-	$\frac{incree}{incree_{input}} < Tol$	$\frac{incree}{F_{fail} - F_{Lebim}} < Tol$	$\frac{incree}{U_{fail} - U_{Lebim}} < Tol$
Incre proposal after acc. failure	$incree_{input}$	$incree_{input}$	$F_{fail} - F_{Lebim}$	$2 \cdot incree_{k-1}$

Furthermore, some figures making use of this code are shown below. They are launched with load control (Figure 4.8) and displacement control (Figure 4.9) and they compare two meshes of element sizes of 0.75mm and 0.0625mm (which is 12 times smaller) for load control and 0.25mm and 0.0625mm for displacement control. These figures show the internal path that the code follows with dashed blue lines,

which consists on the just explained iterative process, and the final path, which refers to the points that are accurate enough according to the tolerance criterion, in red, with relevant points marked with circles.

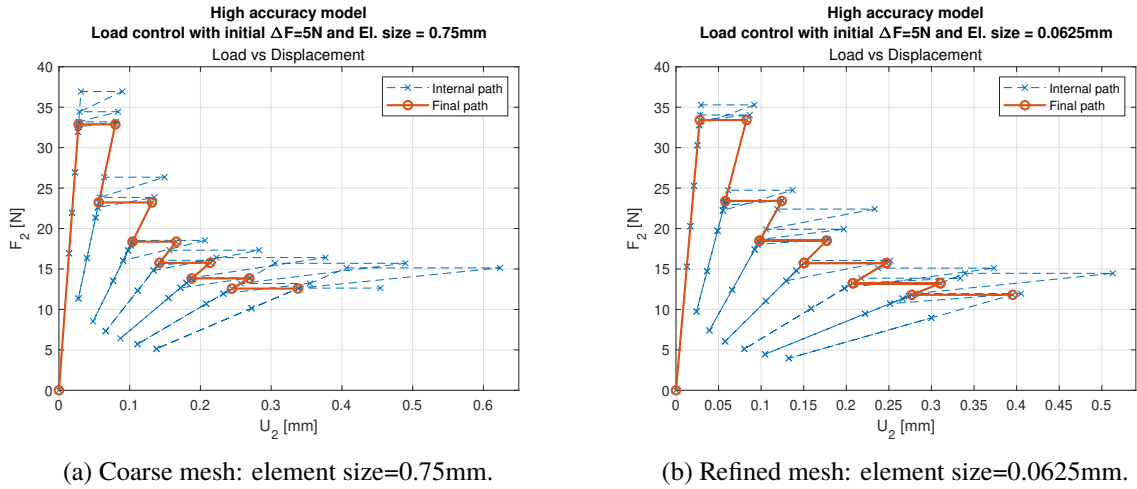


Figure 4.8 High accuracy model - Load control ( $\Delta F = 5N$ ).

In addition, those analyses under displacement control also show the number of damaged elements in each breakage. It is interesting to notice that the chosen tolerance of 10% is enough to perceive the infinitesimal crack growth for the element size of 0.25mm (Figure 4.9a) because each breakage is caused by 1 element failure and it is necessary to increase the displacement to induce the next breakage. On the contrary, for the element size of 0.0625mm, there are cracks of 1 element length but also of 2 and 3 elements. This means that, although the DCB sample (under displacement control) is supposed to have infinitesimal crack growth, smaller load increments are required to catch this effect numerically. For this reason, a parameter that impose infinitesimal growth was implemented for next code version. What this parameter does, if it is selected, is to reduce the increments not only to satisfy the tolerance criterion, but also to ensure cracks of 1 element only.

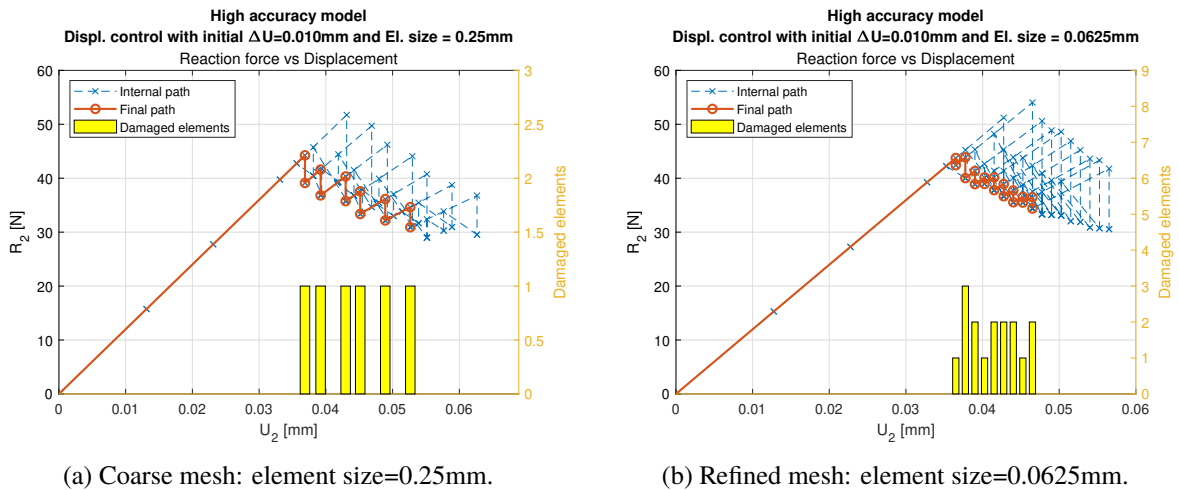


Figure 4.9 High accuracy model - Displacement control ( $\Delta U = 0.01mm$ ).

#### 4.5.2 Final improvements to rise computational efficiency of the codes

This second approach, which has been denominated as adaptative model, consists on an improvement from the previous one. For detailed information about the implementation and how to run the codes,



section C.1 should be consulted.

Anywise, some of the updates are:

- The value that is compared with the tolerance is the last increment with respect to the difference between the last failure load and the *LEBIM load*, which is the required one to initiate damage at one point of the interface according to infinitesimal LEBIM with  $\mu = 1$ .
- It also has a faster loading until reaching the failure by means of rising each increment by a factor (chosen by the user) with respect to the previous one. For example, if this factor is 1.5 (just like it is shown in Figure 4.10), the increments will be increased by 50% each step. This way, in case of choosing a small increment, it will be quickly risen so the number of steps required to reach failure is not too high.
- Furthermore, the proposed increment after each failure is not the one chosen by the user, but chosen by the code with a certain criterion depending on the previous step. For load control, the current proposed increment is  $F_{fail} - F_{LEBIM}$ , whereas for displacement control, the increment will be established as two times the last one, which, for this analysis with infinitesimal growth, has been considered faster than starting again from the input increment. However, this proposals can be modified in the future when more test are known and a deeply knowledge about curves of energetic behaviour allows to improve this approach.
- Some optional parameters, which are specified in the input file, have been also inserted in order to analyse different problems effectively. They are typically binary parameters allowing to activate a certain function or not, or float parameters to indicate a value:
  - `Stop_breakage`: This parameter is 1 when it is wanted to exit from the code when the first failure is reached and 0 if the code continues. This is useful when we are only interested in the knowledge of the failure load, and not in the evolution of the crack. In fact, it will be set on in the adhesive characterization of the DLJ adhesive that will be addressed in section 5.3, where we are only interested in the failure load to be compared with experiments. This way, much computational time calculating subsequent steps will be saved. It is important to remark that this parameter means not continuing with the calculation after failure but it still does the iteration around the breakage according to the tolerance criterion so this load is accurately obtained.
  - `Infinitesimal_fracture`: As previously mentioned, this parameter allows to impose an infinitesimal growth (by giving a value of 1). In this case, once the tolerance criterion is satisfied, the code verifies the number of elements involved in the breakage so if it is greater than one, it reduces the load until reaching this aim.
  - `TOL`: This is the value that will be checked in the tolerance criterion. It should be contained in the range (0, 1] which means from 0% to 100% 'error', so larger values would not have effect and can be set in order to ignore the use of the tolerance criterion and settle for the accuracy of the first calculation.
  - `Stop_under_load`: This parameter was inserted in order to exit the code when a certain load is reached. This is especially useful for problems we know the order of magnitude of the results and unnecessary calculations can be avoided; and to plot different analyses in one figure that reaches the same displacement. Before this parameter, it was only possible to select the number of steps, but depending of the problem, this would involve larger or smaller displacements. With this parameter, graphs like the one showed in Figure 4.13, can be made with analyses with different number of steps but every one reaching the same displacement (0.15mm).
  - `Incre_factor`: This is the factor by which the increment is multiplied every step so the loading until reaching failures is faster. A value that works well is 1.5 (increasing by 50% the value

of the previous step), but this can be set higher for new problems where we do not know the order of magnitude of the failure load; or even 1, which would imply a constant increment in every step. Notice that this parameter only takes place when the internal variable *iterar* is equal to 0, which means that we are in the loading phase; whereas when *iterar=1* we have already reached the failure load and the increments should be reduced to iterate and find the failure point accurately.

- Finally, a launcher script has been developed to be able to perform more than one analysis by changing one parameter. In particular, it has been designed to sweep in  $\tau_{cr}$  and  $G_{II}^{cr}$  so several adhesive properties can be run in series. It has been crucial to carry out the reverse engineering of subsection 5.3.2.

To sum up, this code development has been carried out in order to improve the relationship between computational time and accuracy and to add versatility and allow the user to control the simulations thanks to the insertion of different parameters. In order to clarify the implementation of the iterative process to reach accuracy and the insertion of the just mentioned parameters, a flowchart has been included in Figure 4.10.

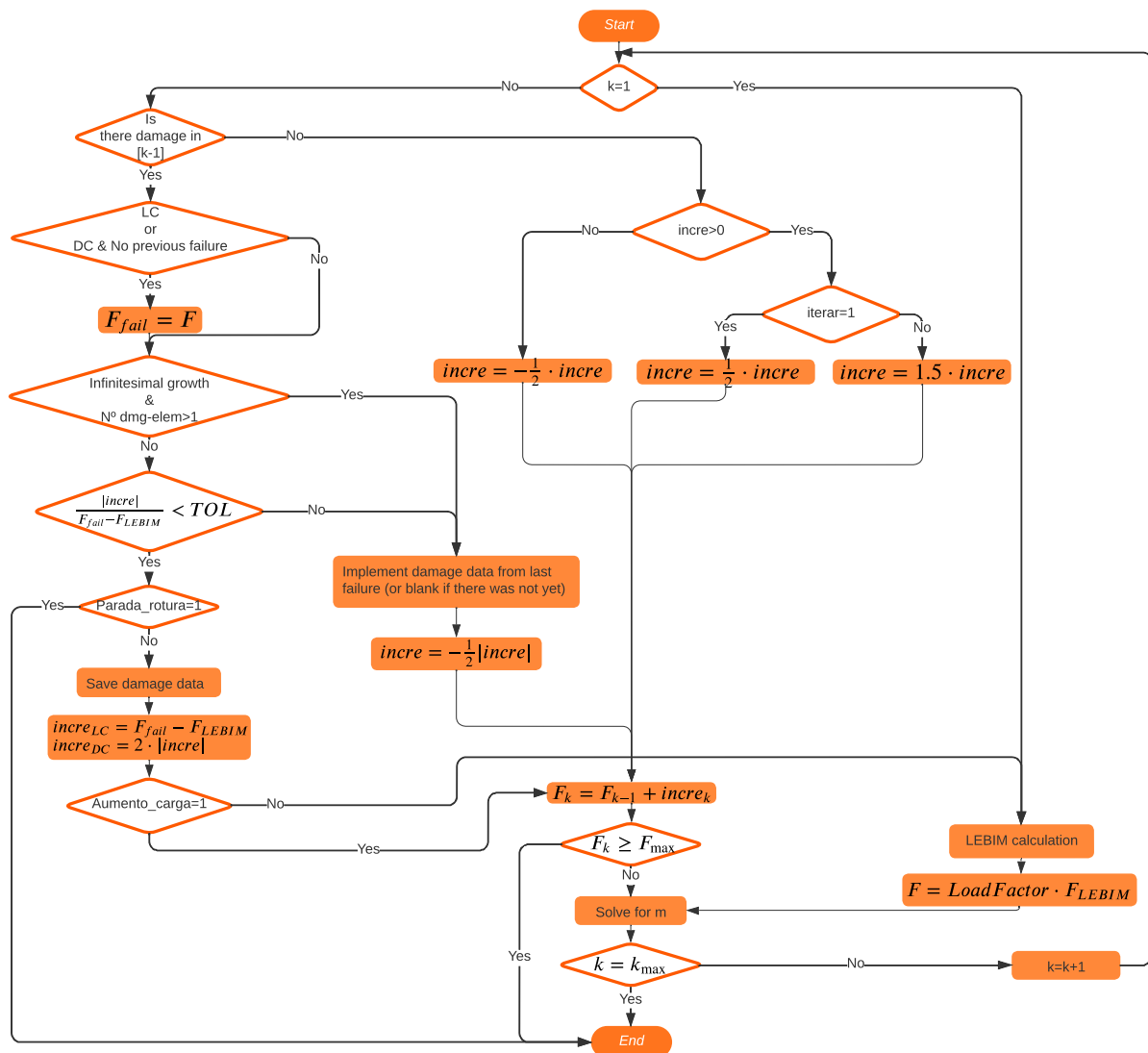


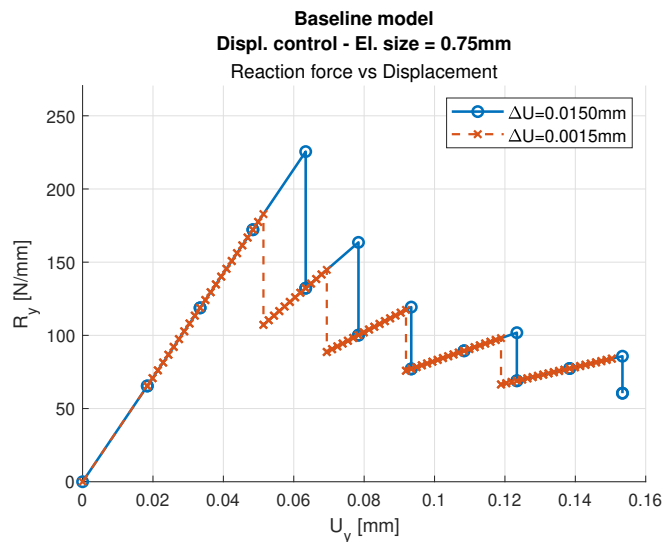
Figure 4.10 Adaptative model flowchart.

It is important to notice that it is focused on the explanation of the implementation and it does not include all the code information. In particular, there are two boxes of meaningful number of lines that are simplified as '*Solve for m*', which includes the couple criteria, and '*LEBIM calculation*' because they are not different from the original code. At the end, the iterative process might be summed up as different branches, which consider the possible cases, ending up in defining an increment so the current step load can be calculated to run the rest of code.

In what is next, some graphs and tables will be shown to compare both models (original and adaptive) with two initial increments in order to see the improvements. Since the conclusions obtained from these analyses are equivalent to those obtained under load control, only displacement control will be mentioned for the sake of brevity.

As we desire to compare load values and computational times, the following analyses have been run up to a displacement of 0.15mm, which is representative enough of the current test.

Figure 4.11 represents the reaction-displacement curve of the test, which has been solved with two different increments that are introduced by the user: 0.015mm in blue and 0.0015mm in red, which is 10 times smaller. It can be seen that the first failure point differs considerably from both approaches. This is somehow expected since, as we previously mentioned, these increments are randomly selected so big ones are inaccurate and small ones are computationally expensive (more than 500%).



**Figure 4.11** Original model - Displacement control.

The results of the just-proposed innovative approach are shown in Figure 4.12, where the comparison between user-introduced increments of 0.0150mm (Figure 4.12a) and 0.0015mm (Figure 4.12b) can be seen. The blue-dashed line represents the internal path the code follows to arrive to the accurate solution, which is pointed out with red line. With respect to the failure load value, it is quite similar between both approaches because we imposed it through the tolerance. In fact, if we reduce this tolerance, both values will be even closer up to the accuracy we desire.

It can also be seen in Figure 4.12 that the required number of steps are not too different between both approaches (32 vs 39) compared to those of the original model (10 to 89). This is thanks to the increment variation of +50% in the loading phase, which results in similar number of steps even if the increment introduced by the user is very small. This can be clearly seen if we compare the number of steps that are required to reach the first failure point for the different models and increments (Table 4.4). It can be appreciated that the adaptive model is more regular and stable so a bad increment choice does not involve too much additional computational cost.

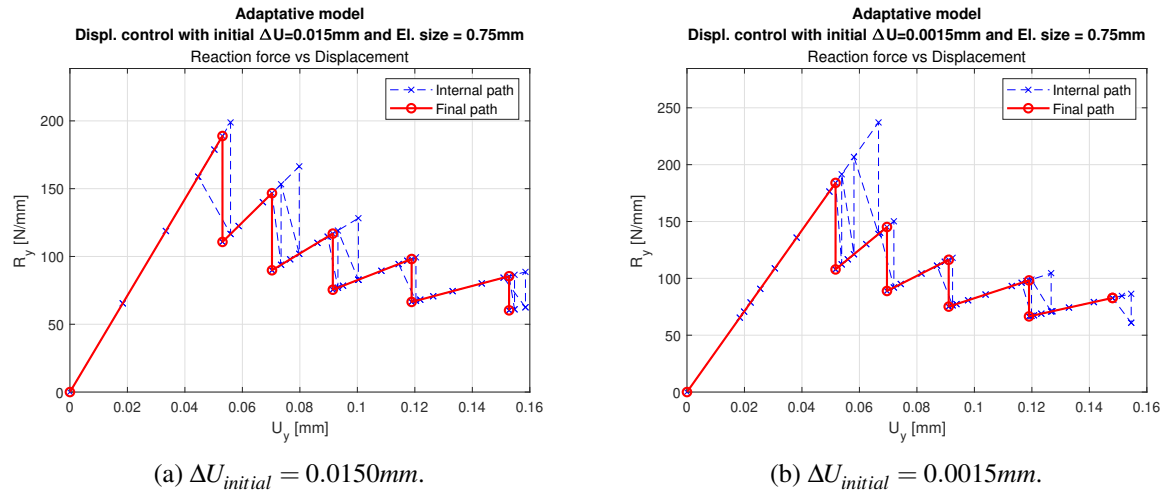


Figure 4.12 Adaptive model - Displacement control.

Table 4.4 Max. load and number of steps required to load until first failure for original and adaptive models.

Model	Initial incr. ( $\Delta U$ ) [mm]	$F_{max}$ [N/mm]	$N_{steps}^o$ to reach failure
Original	0.0150	225.54	3
	0.0015	182.84	22
Adaptive	0.0150	188.85	5
	0.0015	183.94	10

Finally, both models and increment approaches have been simultaneously represented in Figure 4.13 in order to appreciate the differences. The baseline model are represented in green and pink lines for increments of 0.015mm and and 0.0015mm, respectively; and the adaptive model in blue and red lines for increments of 0.015mm and and 0.0015mm, respectively.

In addition, some relevant information such as maximum reaction force and computational costs are detailed in Table 4.5. As it can be appreciated in the original model, if the increment introduced by the user is too big, the accuracy differs a lot. Meanwhile, if it is too small, the computation cost increases considerably. On the other hand, the adaptive model is more regular since large or small increments result on similar accuracy and computational costs. It must be reminded that, for illustrative purposes, the chosen tolerance for these analyses is 10%, although a smaller one can be selected to reach more accurate values.

Table 4.5 Computational cost and number of steps required to reach 0.15mm for original and adaptive models.

Model	Initial incr. ( $\Delta U$ ) [mm]	$N_{steps}^o$	Comput. cost [min]
Original	0.0150	10	14.23
	0.0015	89	86.38
Adaptive	0.0150	32	40.59
	0.0015	39	45.82

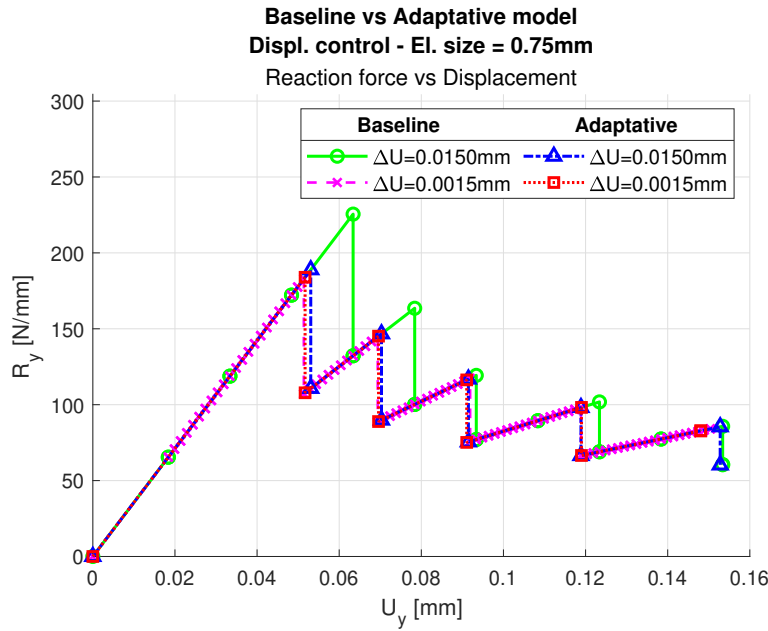


Figure 4.13 Original vs Adaptative model - Displacement control.

## 4.6 Energetic understanding of PMTE-SC applied to DCB

The adaptative control, as it has previously been mentioned, has been designed from a loading point of view, which refers to the observation of graphs of the type Load vs Displacement. This was done because, on the one hand, those graphs were firstly obtained and the necessity to increase accuracy arrived soon; and, on the other hand, the energetic behaviour is unknown in advance so, although we have only observed concave or convex behaviours until now, it is expected that other problems have more complex behaviours. For this reason, it was sensible to focus on the critical failure load. However, it is logic to study the energetic behaviour in order to learn how to implement an adaptative control. In fact, it is the purpose for the near future.

This only means that, although the adaptative process works, at least for the problems considered in this project, the conceptual idea fits better under an energetic control because the PMTE-SC consists on an energetic minimization. This way, the final load obtained by the proposed adaptative code, must be very close to the one that the tool is able to obtain by means of energetic control, but we will never achieve it with 0% error, although it will be close for a small tolerance.

This section is therefore dedicated to deeply understand the energetic implication in the problem by means of the representation and comparison between analytical curves from [14] and the numerical ones.

The following graphs show the dependence of the total energy with respect to the crack length for different load values representing, therefore, several curves. For the analytical curves (Figure 4.14), they magnitudes are dimensionless, whereas for the numerical ones (Figure 4.15), they have dimensions. However, they can be compared because they derive from different problems and we only want to compare the general behaviour and not the values themselves.

The energy that is represented is the total energy, firstly introduced in [11], whose variation is searched to be minimized. This energy is the sum of potential energy ( $\Pi$ ) and dissipated energy ( $R$ ). The potential energy consists of stored strain energy ( $U$ ) for displacement control and stored strain energy minus employed work for load control (Equation 4.1).

$$\Pi = \begin{cases} U & \text{Displacement control} \\ U - W & \text{Load control} \end{cases} \quad (4.1)$$

Both energies depends on  $\Delta a$ , which is the finite crack advance. This way, the energy balance condition that must be fulfilled is  $\Delta\Pi(\Delta a) + \Delta R(\Delta a) \leq 0$  so a crack appears. In fact, the PMTE-SC proposes to minimize this addition (Equation 4.2).

$$\min_{\Delta a < A_{\sigma}} \Delta\Pi(\Delta a) + \Delta R(\Delta a) \quad (4.2)$$

With respect to numerical results (Figure 4.14b and Figure 4.15b), the x-axis represents the number of damaged elements, being the adhesive composed of 80 elements. This was chosen to easily understand the size of the cracks, as well as being able to identify infinitesimal growths, which corresponds to 1 element cracks (not being this the only condition, as it will be seen later).

It is important to remark that the numerical results give us the position of the points, whereas the curves are manually estimated by means of the information they provide and the acquired knowledge of the problem behaviour.

In order to be able to easily understand the graphs, the nomenclature for the markers and line shapes will be detailed:

- Triangles pointing to the right ( $\triangleright$ ) indicate the lower limit of the interval that represents the elements that satisfy the stress criterion. This limit corresponds to the first start of the minimization ( $n=1$ ), which will be better explained later. Since, for the sake of simplicity, there is no previous damage, they are all positioned at the y-axis.
- Triangles pointing to the left ( $\triangleleft$ ) indicate the upper limit of the just mentioned interval, which corresponds to the second start of the minimization ( $n=2$ ).
- Dashed lines represent the curve portion where elements satisfy the stress criterion.
- Circular filled markers ( $\bullet$ ) represent intermediate points of the minimization process. Those points where there is both triangle and circular markers imply they are local minimums, which will be useful for the concave curves representation.
- Quadrangular markers ( $\square$ ) represent the final point of the minimization, which indicates the number of elements that satisfy both stress and energetic criteria and the total energy associated to the final state.

Since CCFFM studies crack initiation and propagation when both stress and energy criteria are fulfilled, the PMTE-SC proposes to determine the elements satisfying the energy criterion from those that already satisfy the stress criterion. From that premise, the energy minimization process is performed from two different starts: considering no damage ( $n=1$ ) and considering that all the elements that satisfy the stress criterion are damaged ( $n=2$ ), which corresponds to the triangle points pointing to the right and left, respectively. These initiations arrive to local minimums. They might coincide, but if they do not, the global minimum (square marker) will correspond to the minimum total energy of both, and it indicates the real crack length. A comment about this issue should be taken into consideration: although these two initiations are enough to caught the behaviour of simple problems where total energy curves are completely convex or concave, for more complex problems it might not be enough. Therefore, the code is prepared to consider more initiations in the minimization process ( $n \geq 2$ ) with intermediate considerations, like damaging half of the elements that satisfy the stress criterion.

#### 4.6.1 Displacement control

Firstly, those analyses under displacement control are represented in Figure 4.14. The increment of the imposed displacement imply higher values for the energy because more strain energy is stored, as well as larger number of elements satisfying the stress criterion. In addition, the energy curves are convex (just like it is demonstrated in [14]). This means that there is only one local minimum, which coincides with the global one; and that the growth is infinitesimal. In both graphs, we appreciate the monotonously

ascending behaviour (positive slope) until a certain displacement value where it begins with horizontal slope. At this displacement, the infinitesimal crack growth starts because for larger values the minimum of the curve will be displaced to the right and the crack would have been appeared before.

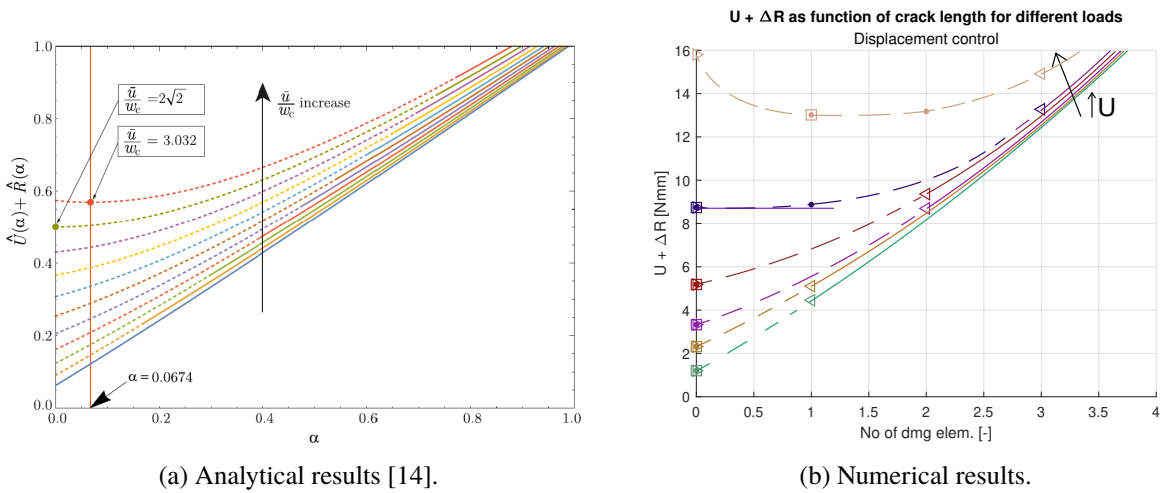


Figure 4.14 Total energy as function of crack length. Displacement control.

For the numerical problem, it is not strictly infinitesimal because elements are finite so they always have a certain size, but there is also a first displacement value at which the crack size corresponds to one element, which would be considered as infinitesimal. If we take a look at the upper curve of the numerical figure, we can see that for that displacement there is one element damaged, however there is a lower displacement that arrives to the same result, which would be represented by a curve between the upper two curves. The objective of the iterative model that has previously been proposed is to be numerically able to calculate a close value to this minimum displacement that induces a crack of one element.

### 4.6.2 Load control

Secondly, those analyses under load control are represented in Figure 4.15.

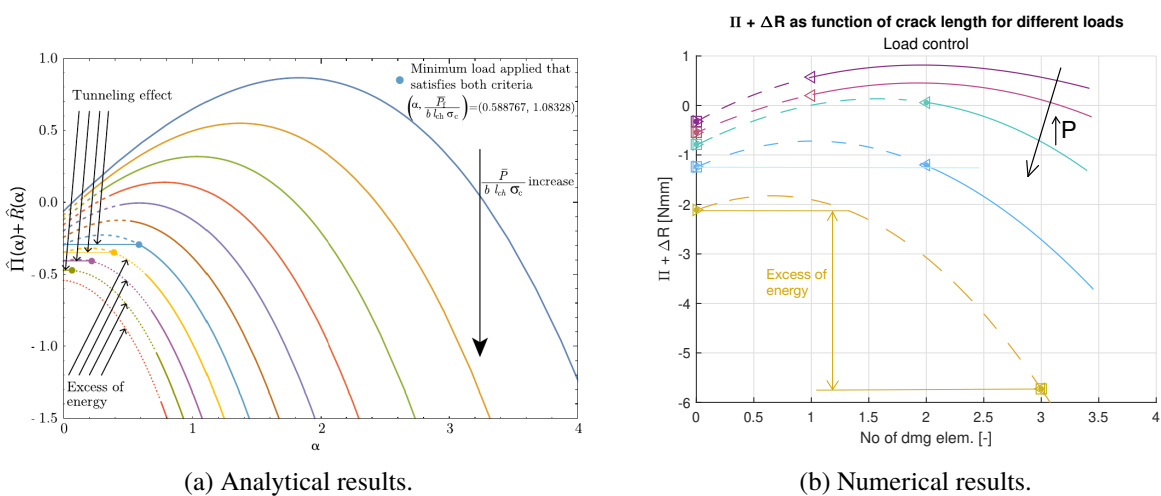


Figure 4.15 Total energy as function of crack length. Load control.

A completely different behaviour is appreciated for several reasons. Firstly, the curves are concave,

which implies a finite fracture. Although it is possible that this crack length is 1 element size, it does not mean that it comes from a conceptually infinitesimal behaviour. In fact, if this happens and we reduce the mesh size, there will be a size by which more than one element breaks even if we adjust the load with tiny increments.

In opposite to what we observed under displacement control, when we increase the load, the energies are smaller. This is because the work increases and, therefore, the potential energy (and total energy) decreases, being able to have a negative value.

In both analytical and numerical graphs, we can see that the number of elements satisfying the stress criterion initially correspond to an ascending part of the curve, where the local and global minimum coincides at crack length of 0 (fuchsia and red numerical curves). Meanwhile, for higher loads, they represent not only ascending curves but descending too (green and blue numerical curves). In this case, although local minimums differ, the global minimum is the first one, so the crack does not initiate under that loads. However, at a certain load (analytical blue curve), both energies coincide (the energy variation is 0) and the tunneling effect appears, inducing a crack of finite length and, since the slope at the curve is negative, the structure collapse with a total crack propagation. For higher loads, the total energy variation is smaller than 0 so the crack is equally propagated but, in this case, there is also an excess of energy.

The adaptative model intends to predict the precise minimum load by which there is no excess of energy, although since this is not the implemented condition, it will never be reached with 0% error. However, we can get very close by iterating with the proposed method to reach the first load that causes the failure. This load will be little higher than the one originating the blue numerical curve.



# 5 Tests: Double Lap Joint (DLJ)

*Genius is one percent inspiration and ninety-nine percent perspiration.*

THOMAS EDISON

Double-lap joints (DLJ in advance) have been typically used to determine the shear strength of epoxy adhesives in applications like aviation. The stress state is known to be composed of peeling and shear stress. However, this complexity implies the necessity to perform tests (in terms of shape and material properties) very similar to their real applications.

## 5.1 Problem description

The DLJ test consists of 3 beams joined by an adhesive, where the external ones are identical so the numerical analyses can be performed under symmetry condition. The free-end of the internal beam is fixed, whereas clamps pull from the free-ends of the external beams to control the load. It is also typical to add any kind of support between the free-ends of the external beams in order to maintain the transversal distance between them because they tend to get closer during the test. This might be numerically modelled with a pin which limits the vertical displacement (see Figure 5.1), although several models will be later proposed and analysed.

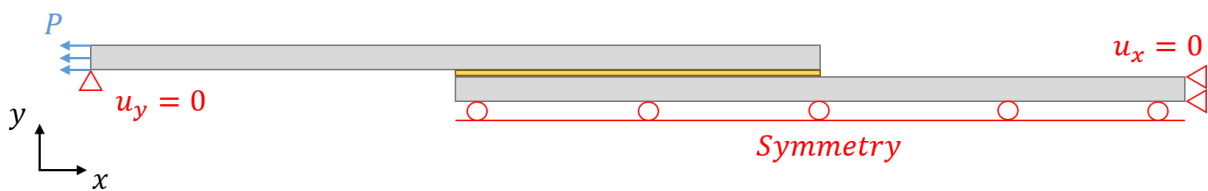


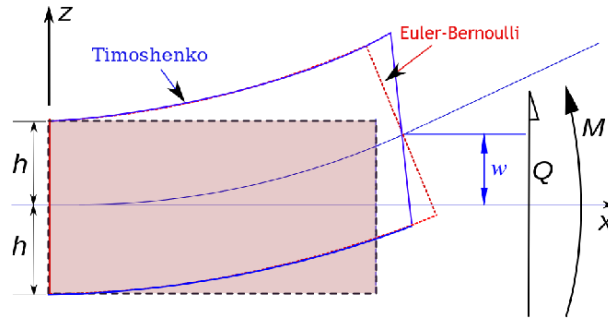
Figure 5.1 DLJ sample.

## 5.2 Comparison between analytical and numerical models

In this section, some comparisons between the proposed numerical model and the analytical formulation developed in [14] for Euler-Bernoulli beams and in section 2.5 of [12] for Timshenko beams.

The main difference between both models is that, in Euler-bernouilli beams, the shear deformation is neglected so any plane perpendicular to the neutral axis before bending will remain so after the beam is bent; whereas the Timoshenko beam takes into consideration the effect of the shear deformation, just like Figure 5.2 shows. The Timoshenko model can be seen as an extension of the Euler-Bernouilli model.

However, the last one still works very well for thin beams, as our problem is. Therefore, similar results are expected from both analytical formulations.



**Figure 5.2** Timoshenko and Euler-Bernoulli beams.

Although the implementation of these analytical models are not the job of this project, some comments of interest will be made. The constitutive equations for the Timoshenko beam are shown in Equation 5.1, which are valid for both isotropic and orthotropic materials, being  $\varepsilon$  the longitudinal membrane strain,  $\chi$  the curvature and  $\gamma$  the shear strain of the beams;  $N$ ,  $V$  and  $M$  the axial force, shear force and bending moment per unit width, respectively;  $E' = \frac{E}{1-\nu^2}$  and  $G$  the elastic modulus for plane strain conditions and in-plane shear modulus, respectively;  $h$  the beam thickness; and  $K_z$  the shear factor of a Timoshenko beam (which will be referred as  $TY$  in the following figures). This factor is  $6/5$  for a rectangular section and, since it allows the possibility to neglect the shear deformation, it will be set 0 to consider the Euler-Bernoulli model.

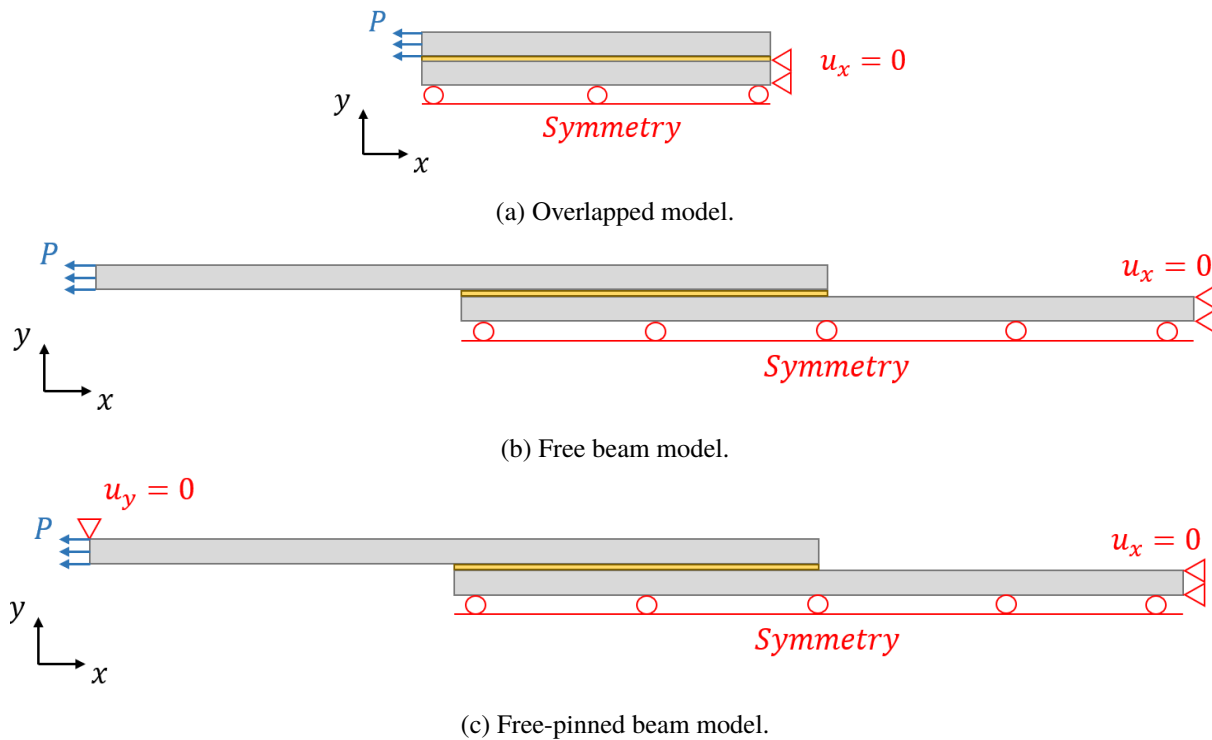
$$\begin{aligned}\varepsilon &= \frac{N}{E'h} \\ \chi &= \frac{12M}{E'h^3} \\ \gamma &= K_z \frac{V}{Gh}\end{aligned}\quad (5.1)$$

Three models (in both analytical and numerical form) will be tested in order to choose the more appropriate one (see Figure 5.3). All of them have an adhesive length of 12.5mm and beam external and internal thicknesses of 1.6mm and 3.2mm, respectively, so  $2 \cdot t_{ext} = t_{int}$ . In the first model, the length of the adherends is 12.5mm, whereas in the last two models it is 25mm, this is, double of the adhesive length. This geometrical characterization of specimens was made in order to replicate the tests from [2], which was the first selection to characterize the adhesive in section 5.3.

- The overlapped model, Figure 5.3a, is the simplest one, which has the minimum needed to model the problem and is easier to analytically implement.
- The free beam model, Figure 5.3b, considers the extension of the beams, being able to caught its behaviour far from the adhesive, since the bending might have influence over it.
- Last, the free-pinned beam model, Figure 5.3c, is similar to the previous one, but including a vertical restriction of the displacement at the free-end of the external adherend, which tries to simulate the real test where the clamps only move in horizontal direction.

Although finally, only aluminium adherends are considered in this project, in order to extend the knowledge for future analyses, both isotropic and orthotropic materials have been tested in this section.

The models are run under static analysis with pressure load of 100 N/mm. Symmetry conditions are applied and the elements employed for adherends and adhesive are CPE4, which are plain strain solid



**Figure 5.3** Numerical models.

elements with 4 nodes. Tie interactions are used to join adhesive and adherends. The adhesive material properties are inserted by means of the \*User Material card to be used in the subroutine:  $\sigma_{cr} = 22.08$ ,  $G_{I,c}^* = 0.024$ ,  $\lambda^{HS} = 0.024375$ ,  $\frac{k_t}{k_n} = 0.2308$ ,  $h_{adhesive} = 0.01$ ,  $\mu = 8.62$ ; all in units congruent for ABAQUS analyses: ton, mm, s, MPa, mJ. For further understanding of the UMAT subroutine and its parameter, [9] should be consulted. The values have been chosen from the literature to be close to real ones but they are not too rigorous because only qualitative behaviours are searched. Anyways, it can be seen that some of them will coincide with the ones used in following sections when the numerical model is intended to be compared with real tests.

### 5.2.1 Isotropic adherends

First of all, Figure 5.4 compares the numerical normal stress ( $S_{11}$  in ABAQUS nomenclature, which is typically known as  $\sigma_{xx}$ ) and shear stress ( $S_{12} \equiv \sigma_{xy}$ ) of the adhesive at upper (solid lines) and lower (dashed lines) node paths in order to be able to identify if there are numerical errors. Different colours are used for normal and shear stress components. The  $x$  and  $y$  axes for the identification of the stress components are shown in Figure 5.3.

It must be noticed that the  $x$ -axis of the figure represents the horizontal distance of the adhesive, which is 12.5mm for the three models. It is appreciated that the maximum stresses are located at the ends, appearing compression at the left and tension at the right due to the bending of the adherends.

From this brief analysis we have first seen the significant influence of both normal and shear stresses, not being possible to neglect none of them, and it is also concluded that no matter where we measure the stresses at the adhesive (upper or lower paths), the numerical results are identical.

In what is next, the analytical and numerical results will be compared. For the isotropic consideration (aluminium with elastic behaviour:  $E = 68670\text{MPa}$  and  $\nu = 0.33$ ), the overlapped model is represented in Figure 5.5 for two element sizes (0.125mm and 0.03125mm, which is 4 times smaller) and for two analytical models (Timoshenko with  $K_z = 6/5$  and Euler-Bernoulli with  $K_z = 0$ ).

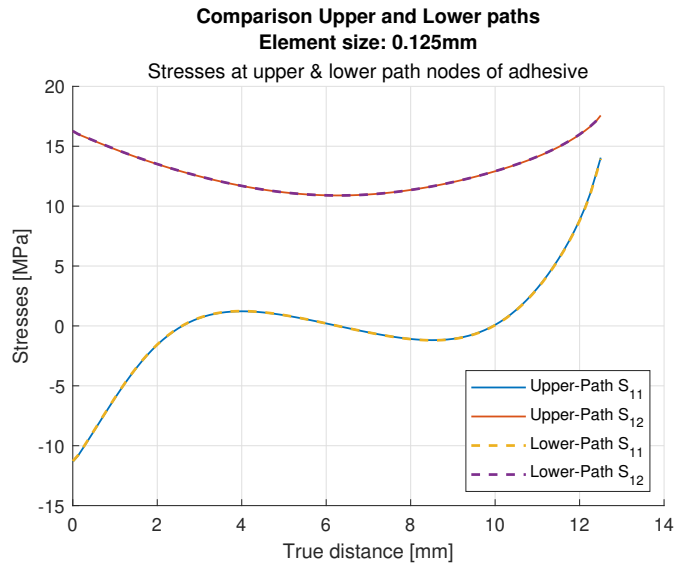


Figure 5.4 Upper and lower stresses path. Normal and shear stresses.

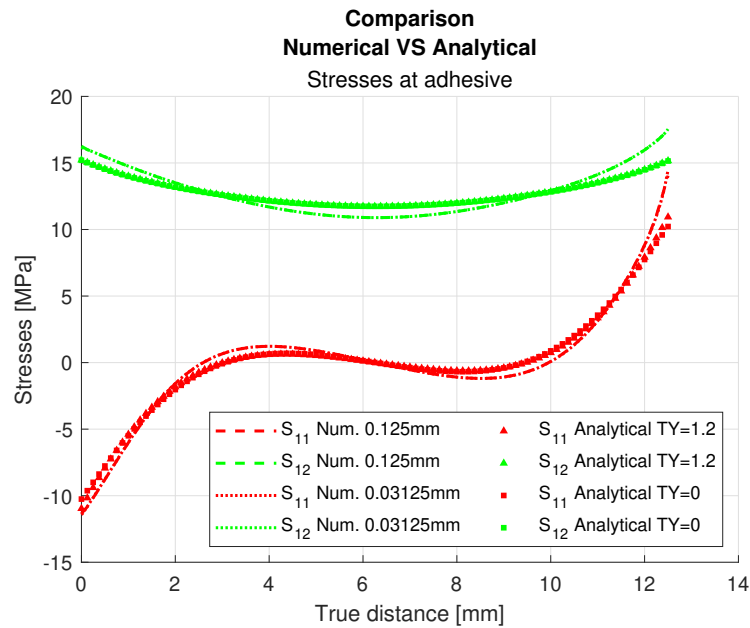
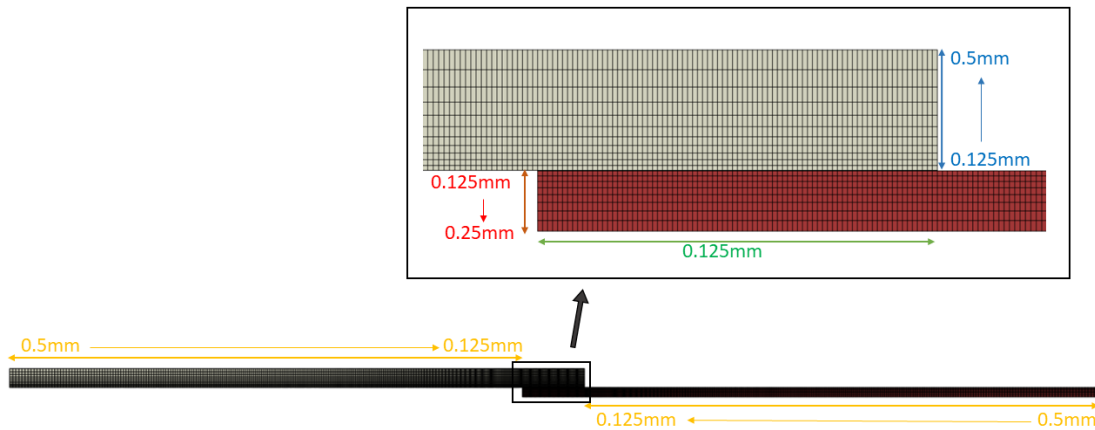


Figure 5.5 Isotropic overlapped model. Numerical (El. size: 0.125mm and 0.03125mm) vs Analytical ( $K_z = TY = 1.2$  and  $K_z = TY = 0$ ).

It is appreciated that identical results are obtained from both mesh sizes of the numerical results. In fact, it is concluded that a finer discretization is done, but the calculation of stresses does not vary. Therefore, the characteristic element length of 0.125mm will be used for future analyses at the adhesive. Furthermore, the complete specimen is shown in Figure 5.6, where element sizes are shown. With the experience of the already presented results it has been concluded that a bias meshing reduces the computational cost without affecting the results. Remember that, although the model is small and simple, it is called and solved many times so this kind of simplifications are convenient. This way, elements of 0.125mm are used for the adhesive and a bias is inserted for the adherends to transition from 0.125mm to 0.5mm for both directions in the external adherend, and from 0.125mm to 0.25mm in internal adherend because it

is half of the external so the symmetry is maintained. The total number of elements is 7395.



**Figure 5.6** Mesh of the specimen for DLJ test.

Coming back to Figure 5.5, symmetry and anti-symmetry results are observed for analytical shear and normal stresses, respectively, because the joint is balanced, which means that  $2 \cdot E_{ext} \cdot t_{ext} = E_{int} \cdot t_{int}$ , where  $t_{int}$  corresponds to the total thickness of the internal adherend so the specimen drawings from Figure 5.3 show half of this thickness due to the symmetry. Meanwhile, some variations appear in the numerical results, specially at the right end. This might be caused due to the boundary conditions and load applied at the beam corners.

On other matters, a good correlation of the results with respect to analytical models from the literature is observed. In particular, very similar distribution of shear and peel stresses are observed compared to those from the DLJ model of Goland and Reissner from 1944 (see Figure 5.7b), which considers the bending of the adherends and peeling of the adhesive. Meanwhile, the model of Volkersen for single lap joints (SLJ) of 1938, which was extended by de Broune in 1944 to DLJ (see Figure 5.7a), only considered axial stresses of the adherends and shear deformations of adhesive. This simpler model is still useful to have a first approach of the shear stresses at the adhesive and normal stresses at the adherends, but is incomplete for further understanding.

Furthermore, it is also observed that shear stress distribution is quite flat. This coincides with the behaviour expected from the model of Volkersen and de Bruyne in 1944 (see Figure 5.8), which predicts flatter distributions for small overlap lengths and high stiffness of adherend with respect to adhesive, which are fulfilled in our specimens.

Anywise, since the analytical analyses are based on a beam model and the numerical one is plane, results are not expected to exactly coincide, although similar behaviours are reached.

Finally, as expected due to the thinness of the beams, no significant differences are appreciated between Timoshenko and Euler-Bernouilli analytical models.

For deeper understanding of these comments and analytical solutions of double lap adhesive-bonded joints, [8] might be consulted.

### 5.2.2 Combination of orthotropic and isotropic adherends

On the other hand, an orthotropic external adherend bonded to an aluminium internal adherend is considered. The composite material properties are compiled from [2], they correspond to carbon epoxy unidirectional long fiber AS4/8552, manufactured by HEXCEL. The mechanical properties are:  $E_{11} = 141.3GPa$ ,  $E_{22} = E_{33} = 9.58GPa$ ,  $G_{12} = G_{13} = 5GPa$ ,  $G_{23} = 3.5GPa$ ,  $\nu_{12} = \nu_{13} = 0.3$  and  $\nu_{23} = 0.32$ , being '1' the fiber direction. In first instance, the section of the composite beam is not modelled as laminate but as *Solid, Homogeneous* in order to simplify.

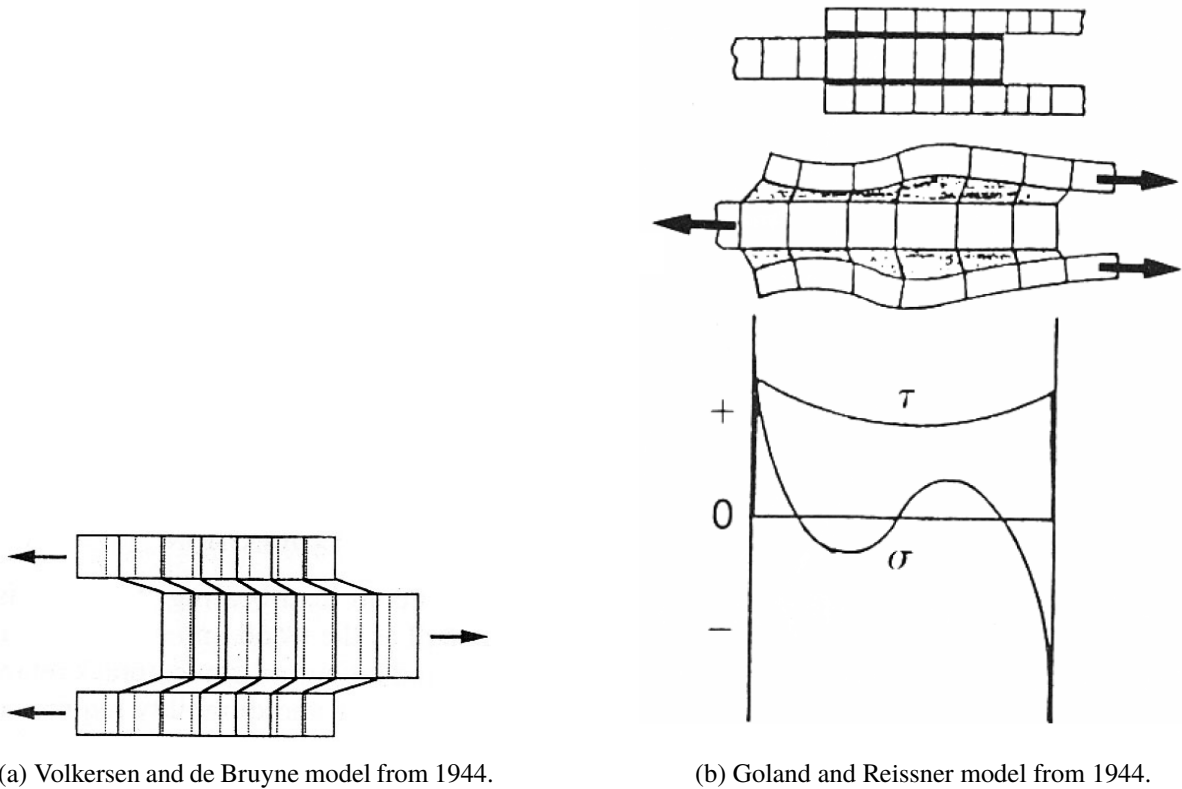


Figure 5.7 First analytical DLJ models. Figures from [2].

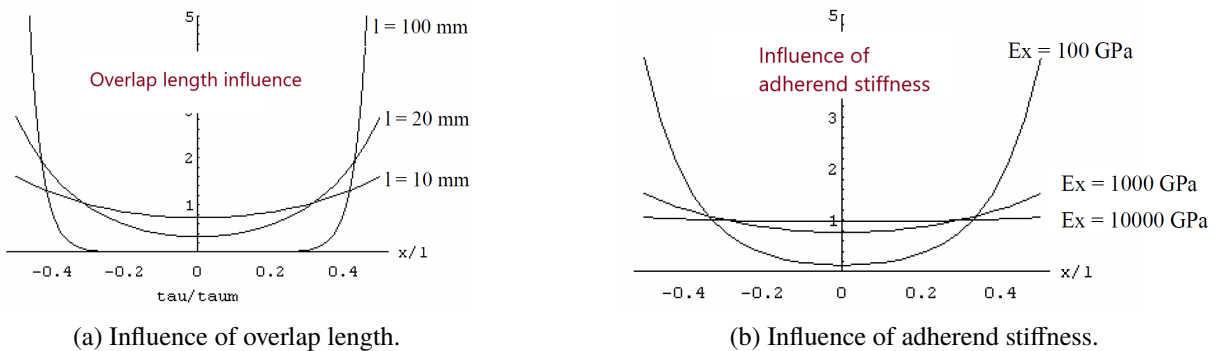
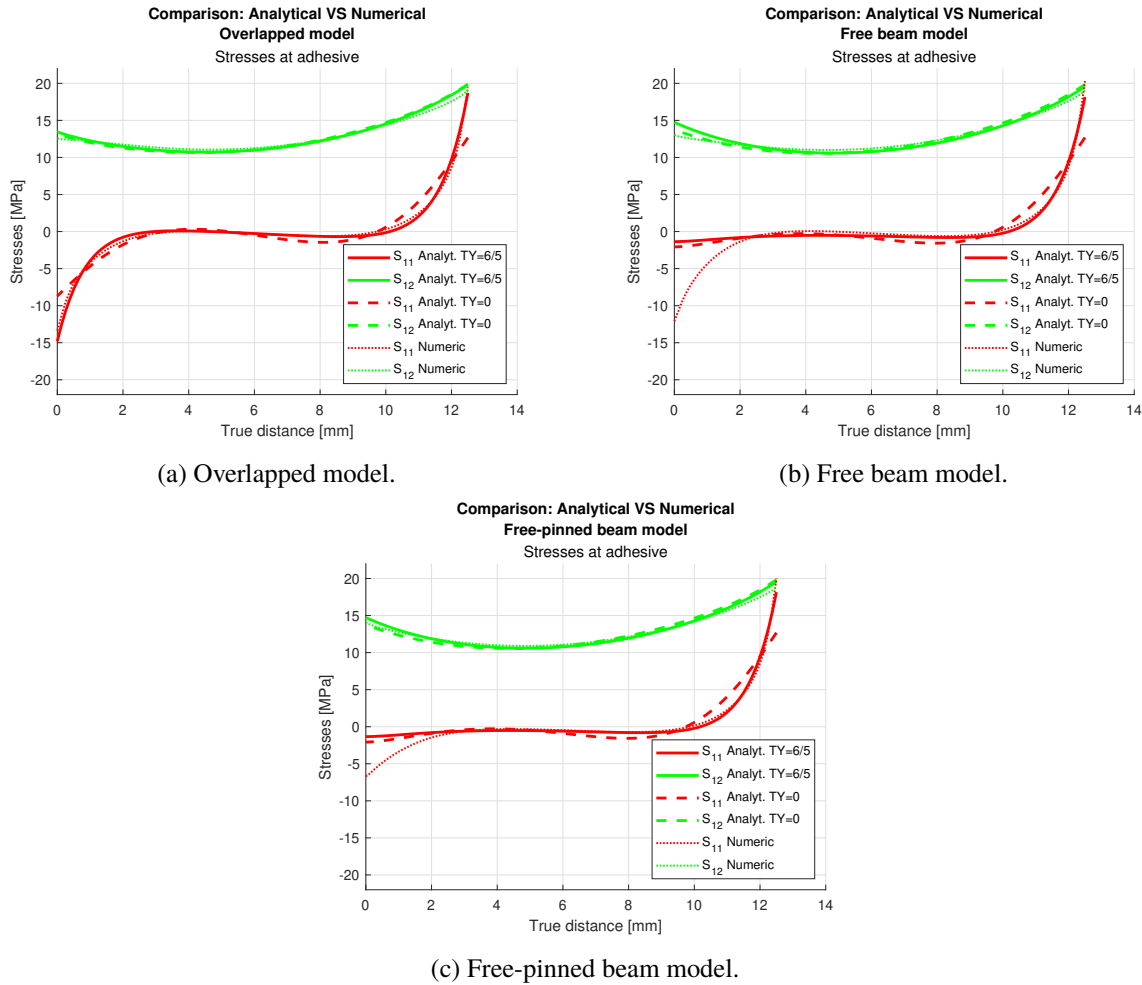


Figure 5.8 DLJ model of Volkersen and de Bruyne in 1944. Parameter influence. (From lesson documentation from A. Barroso).

In Figure 5.9, the results for the three models are shown. Graphs show again numerical results for element size of 0.125mm and analytical results from Timoshenko model ( $K_z = 6/5$ ) and Euler-Bernouilli model ( $K_z = 0$ ).

In this case, since the joint is no longer balanced because, although  $2 \cdot t_{ext} = t_{int}$ , the external adherend stiffness is higher ( $E_{ext} > E_{int}$ ), so as previously mentioned, shear stress peaks appear at the adhesive end where the adherend is less stiff, in this case, at the right.

In addition, there are no big differences between analytical and numerical models in terms of shear stresses, which means that the three models are able to caught this behaviour properly. Furthermore, in terms of peeling stresses, the overlap model show high reciprocity between numerical and analytical results with Timoshenko model ( $K_z = 6/5$ ). However, it is thought that this simpler model does not represent the totality of the specimens which are employed in the real tests that we will later use, where



**Figure 5.9** Orthotropic models. Numerical (0.125mm) vs Analytical ( $K_z = TY = 1.2$  and  $K_z = TY = 0$ ).

adherends are quite longer than the overlap length. For this reason, we will use the third model, which seems to have closer results to those from the analytic curves, for future analyses.

In any case, the general stress behaviour of the models coincides with the one obtained in [19] (see Figure 5.10), where both adherends are made of the same composite material, except for the symmetry that is therein obtained in shear stresses due to the balance of the joint. For that case, the overlap length is  $2c=12.7\text{mm}$ , which is very close to the one we analysed; and beams are lightly thinner than ours ( $2 \cdot t_{ext} = t_{int} = 2\text{mm}$ ).

### 5.3 PMTE-SC. Adhesive characterization

Once we understand the general behaviour of the DLJ, we can implement the Principle of Minimum Total Energy subjected to Stress Condition (PMTE-SC). Firstly, we need to choose tests from the literature and characterize the adhesive LEBIM.

This matter is simplified to the determination of 3 parameters representing stiffness and resistance (in terms of stress and energy) since other parameters can be deduced from them. These parameters can be chosen at both mode I and mode II of fracture, which are also related, so any parameter of each of the following pairs can be used:  $k_n$  or  $k_t$ ,  $\sigma^{cr}$  or  $\tau_{cr}$  and  $G_I^{cr}$  or  $G_{II}^{cr}$ . In our case, since in the problem we are involved the shear stresses are dominant over normal stresses, those parameters related to mode II will be determined:  $k_t$ ,  $\tau_{cr}$  and  $G_{II}^{cr}$ .

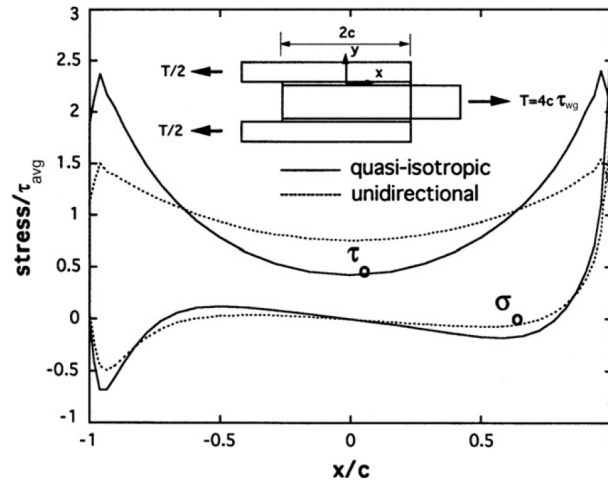


Figure 5.10 Composite adherends from [19]. Shear and peeling stresses.

### 5.3.1 Determination of $k_t$

Firstly, we will estimate the tangential stiffness of the LEBIM springs ( $k_t$ ).

It can be searched in the literature ([2] and [3]) that a typical value for Poisson's ratio of epoxy adhesives is  $\nu = 0.35$ . This way, the relation between tangential and normal stiffness of the LEBIM can be obtained through Equation 5.2 ([20]).

$$\frac{k_t}{k_n} = \frac{1 - 2\nu_l}{2(1 - \nu_l)} = 0.2308 \quad (5.2)$$

This expression tells us that  $0 < \frac{k_t}{k_n} < 0.5$ . In fact, a typical value for this quotient in isotropic interfaces is 0.25 (see [13]).

In first instance, with the purpose of analysing the interaction between aluminium and CFRP adherends due to its importance in the industry, experiments from [2] were used. From those experiments, we selected the ones with 'stripped' finishing (denoted as D due to the Spanish name 'decapado') over the ones with anodized finishing (denoted as A) because, by chance, they experimented a more linear behaviour.

At this point, to achieve our goal by means of reverse engineering we need at least 3 types of tests with a certain changing parameter. The first 3 configurations out of the 11 allowable were selected, where the changing parameter was the thickness of the CFRP adherend. In particular, these configurations consisted of an aluminium internal adherend of 3.2mm and a CFRP external adherend of  $0^\circ$  orientation of 8, 12 and 16 plies.

For each configuration, 4 tests were done in order to reduce scattering. Those obtained for the first configuration (Test 1D) are shown in Figure 5.11.

In addition, the experiments were computerized by reading the colour of the pixels to be able to represent them together in MATLAB and with additional data. Figure 5.12 shows them altogether and with an additional green-dashed line which represents the mean of the experiments by assuming a linear behaviour between the origin (point [0,0]) and the point at which ultimate force and displacement are reached.

Figure 5.12 shows that the behaviour of the specimen is not linear, especially at the beginning, which might be caused by an initial slipping of the clamps since the stiffness cannot be naturally increased. This event complicates our purpose of determining the stiffness, although it is understood that these experiments were valid in [2] because their purpose was to determine the ultimate forces.

After observing the discrepancy between the expected linear behaviour and the experiments and, although the stiffness could be approximately obtained from the second half of the loading where the be-



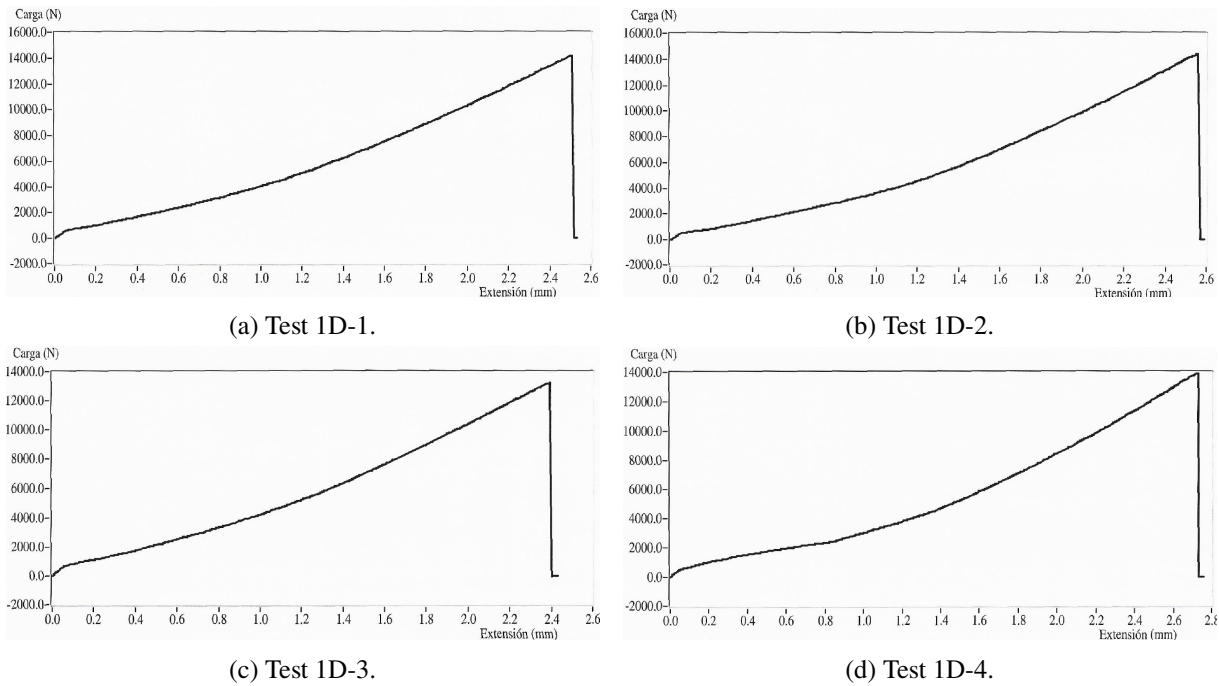


Figure 5.11 Experiments from [2]: Tests 1D.

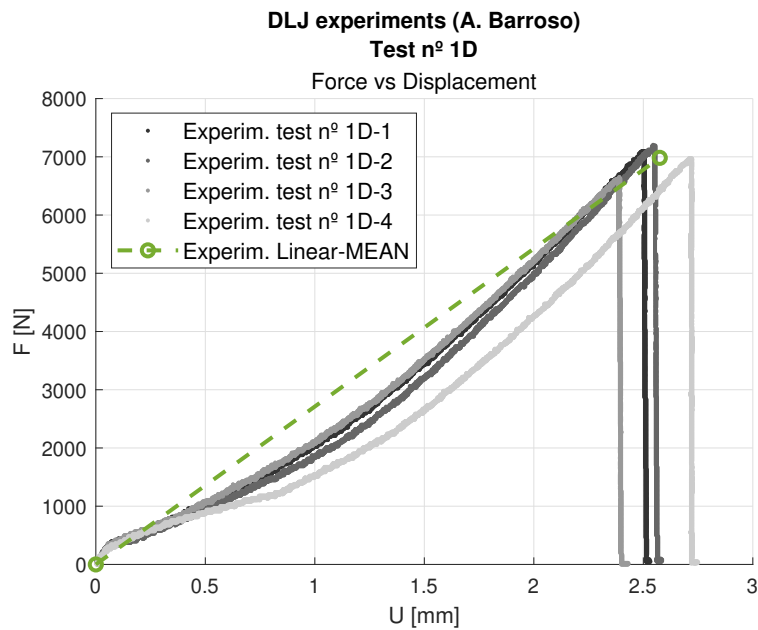


Figure 5.12 Compilation of experiments from A. Barroso ([2]): Test 1D .

haviour looks much more linear, it has been concluded that other experiments should be searched to determine  $k_f$ .

In this line, since appropriate experiments with CFRP external adherends were not found, those from [3] were considered good enough (in the sense of linear response) to characterize our adhesive, although aluminium adherends were employed. In particular, they are made of aluminium alloy AW6082 T651, which is characterized by a high tensile strength of  $340\text{MPa}$ , a Young's modulus of  $E = 70.07 \pm 0.83\text{GPa}$ , a yield stress of  $\sigma_y = 261.76 \pm 7.65\text{MPa}$ , a maximum strength of  $\sigma_{max} = 324 \pm 0.16\text{MPa}$  and a failure strain of  $\varepsilon_f = 21.70 \pm 4.24\%$ . On the other hand, the adhesive employed is Araldite AV138, whose

properties are detailed in Figure 5.13.

Property	AV138
Young's modulus, $E$ (GPa)	$4.89 \pm 0.81$
Poisson's ratio, $\nu^a$	0.35
Tensile yield strength, $\sigma_y$ (MPa)	$36.49 \pm 2.47$
Tensile failure strength, $\sigma_f$ (MPa)	$39.45 \pm 3.18$
Tensile failure strain, $\varepsilon_f$ (%)	$1.21 \pm 0.10$
Shear modulus, $G$ (GPa)	$1.56 \pm 0.01$
Shear yield strength, $\tau_y$ (MPa)	$25.1 \pm 0.33$
Shear failure strength, $\tau_f$ (MPa)	$30.2 \pm 0.40$
Shear failure strain, $\gamma_f$ (%)	$7.8 \pm 0.7$

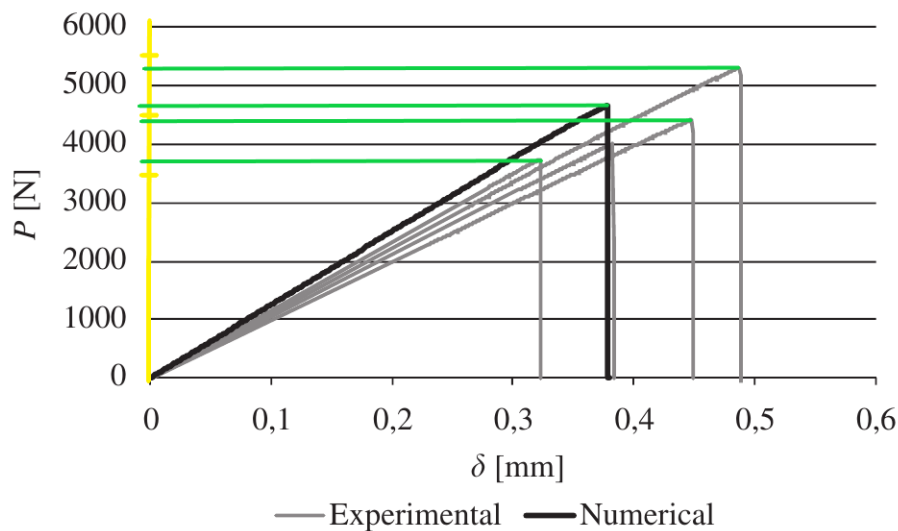
<sup>a</sup> Manufacturer's data.

**Figure 5.13** Properties for adhesive Araldite AV138 [3].

The joint geometry is as follows: adherent thickness of 3mm, adhesive thickness of 0.2mm, joint total length between grips of 180mm and an overlap length of 5,10 and 20mm, for each of the 3 different cases. The one with overlap length of 10mm is already shown in Figure 5.6. Tests were carried out in a Shimadzu AG-X 100 testing machine with a 100kN load cell.

The load-displacement curves for the experiments with overlap length of 5 and 20mm are represented in Figure 5.14 and Figure 5.18, respectively. As it can be appreciated, there is a loss in the stiffness of the second configuration, which is produced by the plasticity of the specimens; whereas the curves for the first configuration look much more linear. For this reason, in order to characterize  $k_t$ , the curves for the overlap length of 5mm will be selected.

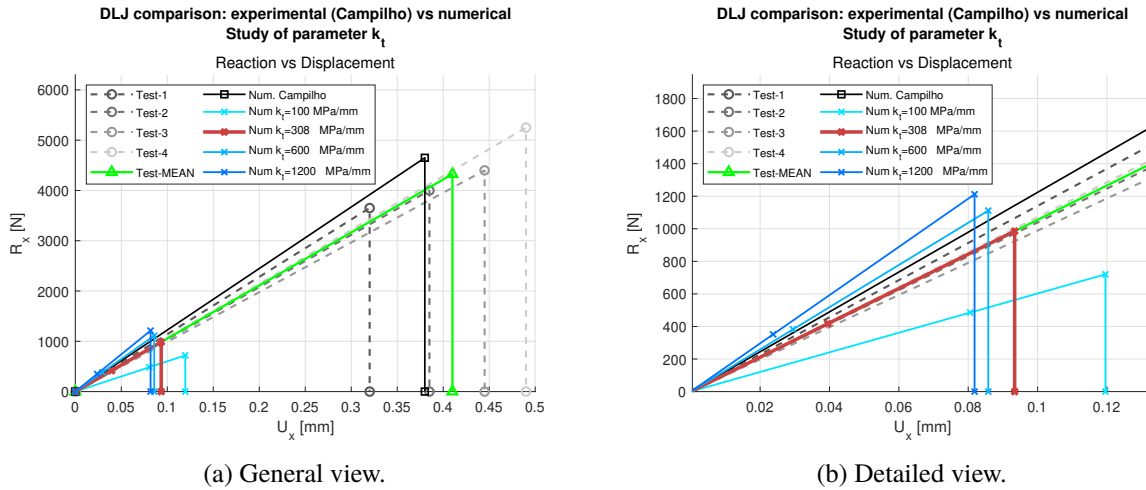
This plasticity effect for large overlap lengths does not concern us because, although failure is expected to be cohesive (across the adhesive) so the adhesive characterization is crucial, linear experiments are only needed to determine  $k_t$ . Meanwhile, fracture parameters like  $G_{II}^{cr}$  and  $\tau_{cr}$  do not depend on  $k_t$  and, therefore, curves from specimens which have suffered plasticity can also be taken as reference.



**Figure 5.14** Experiments for overlap length of 5mm [3].

Firstly, the relevant points of the curves from experiments (Figure 5.14) have been estimated from the graph to be able to linearly represent them in MATLAB with grey-dashed lines in Figure 5.15, where loads over displacements at clamps have been represented. Then, a mean curve (green line) has been

obtained from the 4 experiments. In addition, the numerical result of Campilho have been represented in black-solid line. Last, the just-developed numerical tool has been employed under displacement control (as in experiments) to determine the blue and red solid curves. An iterative process (blue-solid lines) has been carried out to determine  $k_t$  looking for slopes coincidence. We have started with an estimation of  $k_t$  of  $1200\text{MPa}/\text{mm}$  up to a final value of  $k_t = 308\text{MPa}/\text{mm}$  (red-solid curve), which fits with the slope of the mean curve of the experiments.



**Figure 5.15** Determination of parameter  $k_t$  from numerical tool and experiments [3].

It is important to point out that, as Figure 5.15 shows, the order of magnitude of the failure force and displacement for numerical results with respect to experiments are from 3 to 5 times lower because the parameters affecting this event ( $G_{II}^{cr}$  and  $\tau_{cr}$ ) have only been estimated since they are not relevant before  $k_t$  is determined. In particular, the properties employed for these analyses are  $G_{II}^{cr} = 0.42\text{ N/mm}$  and  $\tau_{cr} = 5.8\text{ MPa}$ , which comes from [13], just like Table 5.1 shows for the specimens of that project. However, Double Pull-Push Shear test (DPPS) and DLJ experiments are quite different so properties are only taken from them to have a first order of magnitude.

**Table 5.1** Interface properties for different tests used in [13].

Specimens	$k_t$ [MPa/mm]	$\frac{k_t}{k_n}$	$G_{IIc}$ [N/mm]	$\tau_c$ [MPa]	$\tau_{max}$ [MPa]	$\frac{\tau_c}{\tau_{max}}$
30B	280	0.25	0.42	5.8	15.34	0.38
60C	90	0.25	0.42	5.8	8.69	0.67
90B	340	0.25	0.42	5.8	16.90	0.34
120B	65	0.25	0.42	5.8	7.39	0.78

In addition, Table 5.2 describes the parameters that has been used in the final analysis, where the tangential stiffness is determined as  $308\text{ MPa}/\text{mm}$ . The first five rows corresponds to the input (independent) parameters, whereas the other (dependent) parameters are obtained from the initial ones. The fourth column indicates the position to be read by the UMAT subroutine at those properties that are included in the *.inp* file to characterize the adhesive. It must be noticed the small value of the thickness ( $h$ ), which is needed for the LEBIM to work properly because it neglects the normal stresses in the load direction ( $\sigma_2$ ). Although adhesives are thicker than  $0.01\text{ mm}$  (typically  $\sim 0.2\text{ mm}$ ), they are small compared to adherends and this approximation can be done.

Table 5.2 Interface properties.

Parameter	Units	Description	UMAT pos.	Value
$h$	mm	Interface thickness	5	0.01
$k_t$	MPa/mm	Tangential stiffness	-	308.0
$\frac{k_t}{k_n}$	-	Stiffness relation	4	0.2308
$\tau_{cr}$	MPa	Critical tangential stress	-	5.8
$G_{II}^{cr}$	N/mm	Fracture toughness (Mode II)	-	0.42
$k_n$	MPa/mm	Normal stiffness	-	1334
$\lambda^{HS}$	-	Hutchinson and Suo parameter	3	0.5
$G_I^{cr}$	N/mm	Fracture toughness (Mode I)	-	0.21
$\sigma^{max}$	MPa	Max. normal stress	-	23.67
$\tau_{max}$	MPa	Max. tang. stress	-	16.08
$\mu$	-	Interface brittleness param.	6	7.691
$\sigma^{cr}$	MPa	Critical normal stress	1	8.537
$G_I^{cr*}$	N/mm	Energy associated to stress criterion ( $\sigma^{cr}$ )	2	0.02731

### 5.3.2 Determination of $G_{II}^{cr}$ and $\tau_{cr}$ . Reverse engineering

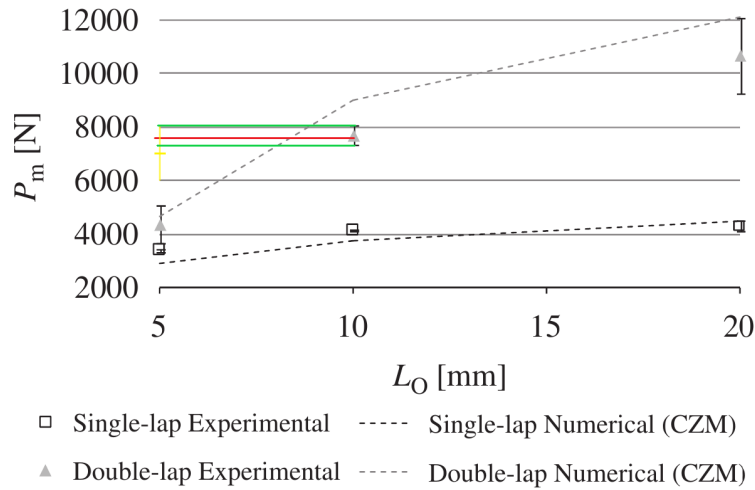
After  $k_t$  is obtained, we will determine  $G_{II}^{cr}$  and  $\tau_{cr}$ . Although it would be more convenient to characterize the interphase through tests designed to determine these properties, the lack of them forces us to use other methods. In this case, we will attempt to characterize the interphase through reverse engineering, also known as backwards engineering. This method is typically used to characterize methods through deductive reasoning to adjust their solution to an existing product with already known results. This process consists on 'trying' different parameters until reaching the desired results, which will indicate the appropriate ones have been chosen. Some additional examples and details about reverse engineering can be found in [21] and [13].

In this project, the reverse engineering will be carried out through numerical analyses. In particular, we will fix the already obtained parameter  $k_t = 308 \text{MPa/mm}$  and we will iterate for the other two parameters of interest ( $G_{II}^{cr}$  and  $\tau_{cr}$ ) until they induce results that adjust to those from the literature ([3]), which are shown in Figure 5.16. This must be done for at least 3 similar tests in which a parameter varies. In our case, this parameter is the overlap length, which will be 5, 10 and 20mm, according to those values from the experiments. The Figure 5.16 represents the failure loads for the three overlap lengths that are considered in [3]. Experimental results are shown with vertical lines that represent the range of the different tests and marks for the means: triangles for double-lap joints (which we are interested in) and squares for single-lap joints. The dashed lines of cohesive zone models (CZM) are not relevant for this project.

In order to be able to numerically implement the reverse engineering, an additional script was developed to sweep in the just mentioned parameters (see Appendix A). This script runs the already developed model for each combination of the parameters with the objective of determining a failure load for every pair of parameters,  $F_{failure} = f(G_{II}^{cr}, \tau_{cr}, k_t = 308 \text{MPa/mm})$ . This way, after solving for an appropriate range of both parameters we will be able to interpolate a failure surface from the obtained points. From the intersection of this surface and the horizontal planes that represent the failure loads from the experiments we will obtain curves which indicate, for each test, the combination of the parameters that would provoke the experimental failure load. When we do this for at least 3 different tests, the curves will theoretically converge into a point (which will be a region in reality) where we will be able to determine the value for  $G_{II}^{cr}, \tau_{cr}$  that allow to obtain failure loads similar to those from the experiments for every type of test. At this point, the problem will be closed.

In order to have a first estimation of the range of the parameters, two approaches will be carried out.

Firstly, the critical shear stress will be defined as function of the fracture toughness and tangential



**Figure 5.16** Experiments for overlap length of 5mm, 10mm and 20mm [3].

stiffness by means of the maximum stress ( $\tau_{max}$ ), which is defined as  $\tau_{max} = \sqrt{2 \cdot k_t \cdot G_{II}^{cr}}$ . This dependence will be estimated from the literature. In particular, by taking a look at Table 5.1, whose values come from [13], we can approximate the relation between  $\tau_{cr}$  and  $\tau_{max}$  as  $\frac{\tau_{cr}}{\tau_{max}} = 35\%$ , since our  $k_t = 308$  MP/mm.

In this first approach, which is represented in Figure 5.17a, we have increased  $G_{II}^{cr}$  from 0.42 up to a value close to the desired point (where the mean of the tests breaks). The values of  $G_{II}^{cr}$  used for those analyses and the corresponding  $\tau_{cr} = 0.35 \cdot \tau_{max}(G_{II}^{cr})$  are detailed in Table 5.3.

**Table 5.3** Critical parameters of first approach.

Numerical analysis No.	$G_{II}^{cr}$ [N/mm]	$\tau_{cr}$ [MPa]
1	0.42	5.63
2	1	8.69
3	5	19.42
4	8	24.57

By observing the just mentioned graph, we can estimate a range of the fracture toughness. However, in order to understand the impact of the critical stress value, the second approach was introduced. In particular, we analysed for two different  $G_{II}^{cr}$  (5 and 8 N/mm), the influence of two  $\tau_{cr}$  (20 and 40 MPa). The results can be seen in Figure 5.17b.

What can be deduced from both graphs of Figure 5.17 is that the fracture toughness has a greater influence in the appearance of failure, although  $\tau_{cr}$  also affects this point. Therefore, a range for those parameters of  $G_{II}^{cr} \in [5, 8]$  and  $\tau_{cr} \in [20, 40]$  would be enough. However, taking into account that this order of magnitude comes from only one kind of test (overlap of 5mm), a wider range is going to be used to be able to caught the behaviour of the three analyses that are considered (overlaps of 5, 10 and 20mm), just in case they differ from the predictions we have just obtained. Therefore, the final range selected for the numerical interpolation of  $G_{II}^{cr}$  and  $\tau_{cr}$  are [3,14] and [14,45], respectively. In particular, 9 and 8 equidistant values for each range, respectively, will be considered (Equation 5.3):

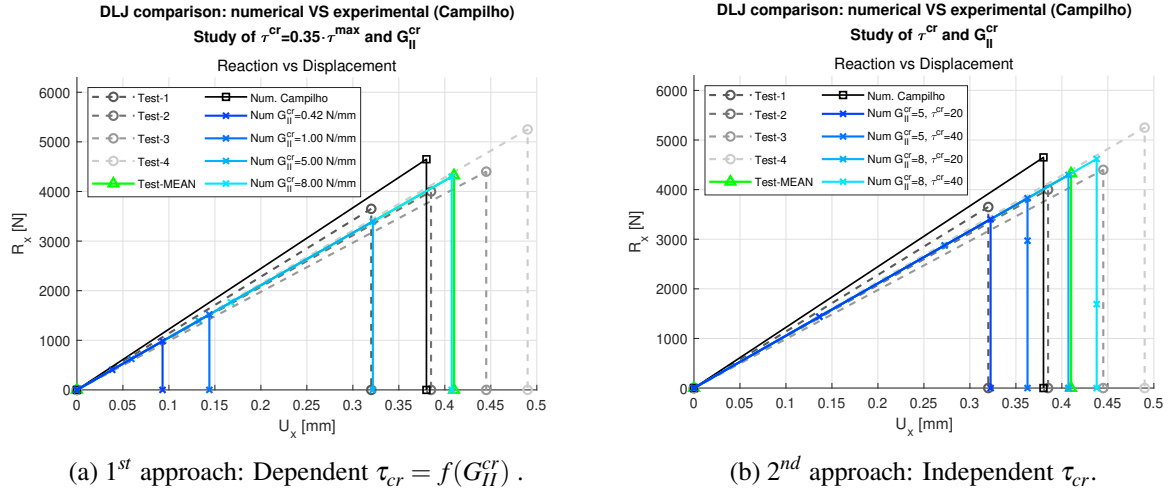


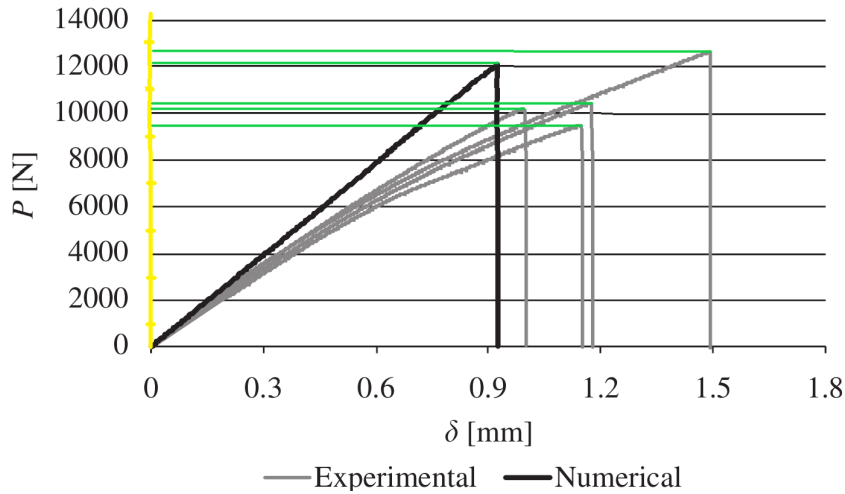
Figure 5.17 Estimation of parameters  $G_{II}^{cr}$  and  $\tau_{cr}$  [3].

$$G_{II}^{cr} = \begin{bmatrix} 3 \\ 4.375 \\ 5.75 \\ 7.125 \\ 8.5 \\ 9.875 \\ 11.25 \\ 12.625 \\ 14 \end{bmatrix} \text{ N/mm} \quad \tau_{cr} = \begin{bmatrix} 14 \\ 18.43 \\ 22.86 \\ 27.29 \\ 31.71 \\ 36.14 \\ 40.57 \\ 45 \end{bmatrix} \text{ MPa} \quad (5.3)$$

The combination of all parameters involves 72 numerical resolutions for each of the three test. At this point, a consideration should be mentioned: all the combinations are not valid because  $\tau_{cr}$  should be smaller than  $\tau_{max}$  ( $\tau_{cr} < \tau_{max}$ ). This way, for a fixed  $k_t$ , large  $\tau_{cr}$  are not possible for low  $G_{II}^{cr}$ . In particular, for  $G_{II}^{cr} = 3$  N/mm, the corresponding  $\tau_{max}$  is 42.99, which is smaller than the last proposed value (45 MPa). Therefore, the range of  $\tau_{cr}$  for the analyses with the smallest  $G_{II}^{cr} = 3$  is [14,38.69], being the upper limit a 90% of the just mentioned  $\tau_{max}$ .

Once the ranges are chosen, we must determine the failure surfaces that will be compared to the numerical results. From [3], Figure 5.14, Figure 5.18 and Figure 5.16, have been obtained. The first and second ones, which correspond to overlap lengths of 5 and 20 mm, respectively, show the data for the 4 tests that were used to reduce scattering, whereas the third one is a summary of all the tests of the 3 kind of experiments, including maximum, minimum and mean values. Therefore, the failure loads corresponding to the 10mm overlap test, that are obtained from the graph of Figure 5.16, are less precise than those from overlap lengths of 5 and 20mm. The approximated failure loads, in [N], have been compiled in Table 5.4.

The procedure continues with the interpolation of the numerical results by means of the *Curve Fitter* tool of MATLAB. In order to smooth a surface like the one showed in Figure 5.21, the tool needs 3 vectors representing the 3 coordinates of each point ( $G_{II}^{cr}$ ,  $\tau_{cr}$  and  $F^{fail}$ ). An explanation about these vectors will be made because the way the tool need the input is not trivially known. For a certain type of overlap analysis, the vectors contain 72 components because we have 72 numerical results. The objective is to define them so each component represents a point. Therefore, since for each  $G_{II}^{cr}$ , we loop in  $\tau_{cr}$ . The  $G_{II}^{cr-vect}$  will consist of each value of the original vector from Equation 5.3 repeated 8 times consecutively; whereas  $\tau_{vect}^{cr}$  will be made of 9 times repetition of the whole original vector from Equation 5.3, except for the first one, which was different due to the limitation of  $\tau_{max}$ . As it is expected,  $F_{vect}^{fail}$  is composed of 72 failure loads of the corresponding combination of  $G_{II}^{cr}$  and  $\tau_{max}$ . An example for these vectors is shown



**Figure 5.18** Experiments for overlap length of 20mm [3].

**Table 5.4** Failure loads [N] for tests from [3].

	Overlap length		
	5mm	10mm	20mm
Test 1 (min)	3650	7320	9500
Test 2	4000	-	10100
Test 3	4400	-	10400
Test 4 (max)	5250	8100	12700
Mean	4325	7640	10675

in Equation 5.4, where the failure loads correspond to analyses with an overlap length of 5mm.

$$\begin{aligned}
 G_{II-vect}^{cr} &= \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ \vdots \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \\ 14 \end{bmatrix} N/mm & \quad \tau_{vect}^{cr} &= \begin{bmatrix} 14 \\ 17.53 \\ 21.05 \\ 24.58 \\ 28.11 \\ 31.64 \\ 35.16 \\ 38.69 \\ \vdots \\ 14 \\ 18.43 \\ 22.86 \\ 27.29 \\ 31.71 \\ 36.14 \\ 40.57 \\ 45 \end{bmatrix} MPa & \quad F_{vect}^{fail} &= \begin{bmatrix} F^{fail}(G_{II-vect}^{cr}(1), \tau_{vect}^{cr}(1)) \\ F^{fail}(G_{II-vect}^{cr}(2), \tau_{vect}^{cr}(2)) \\ \vdots \\ F^{fail}(G_{II-vect}^{cr}(71), \tau_{vect}^{cr}(71)) \\ F^{fail}(G_{II-vect}^{cr}(72), \tau_{vect}^{cr}(72)) \end{bmatrix} = \begin{bmatrix} 2626 \\ 2641 \\ \vdots \\ 5683 \\ 5676 \end{bmatrix} N
 \end{aligned} \tag{5.4}$$

The type of interpolation that has been employed is the *Lowess* one, which is specially designed for surfaces, with the *LowessFit* method, which cannot be selected from the *Curve Fitter* tool but it can be



done in the function. Figure 5.19 shows how the tool looks once the input data is introduced. As it does not allow to use all the methods and options from the interface, it is recommended to use it directly from scripts with the help of MATLAB documentation once it is understood.

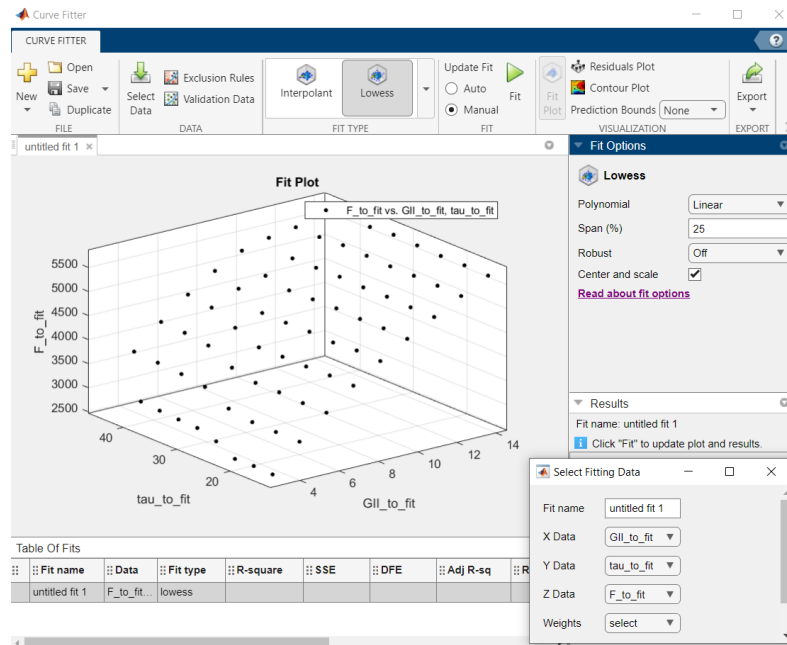


Figure 5.19 Curve Fitter MATLAB tool.

The span percentage is the error that we allow in the interpolation between the resulting surface and the input points. For low span values, the surface will be very close to the input points, but it could be sharpened far from them. On the contrary, for larger values of the span, the surface will be farther from the original points but it will be smoother. In case we were interested in points of the surface far from our range of values, it is crucial that the surface was smooth (large value of span). Otherwise, the surface interpolation will be very distorted from the real one, at least for this interpolation method.

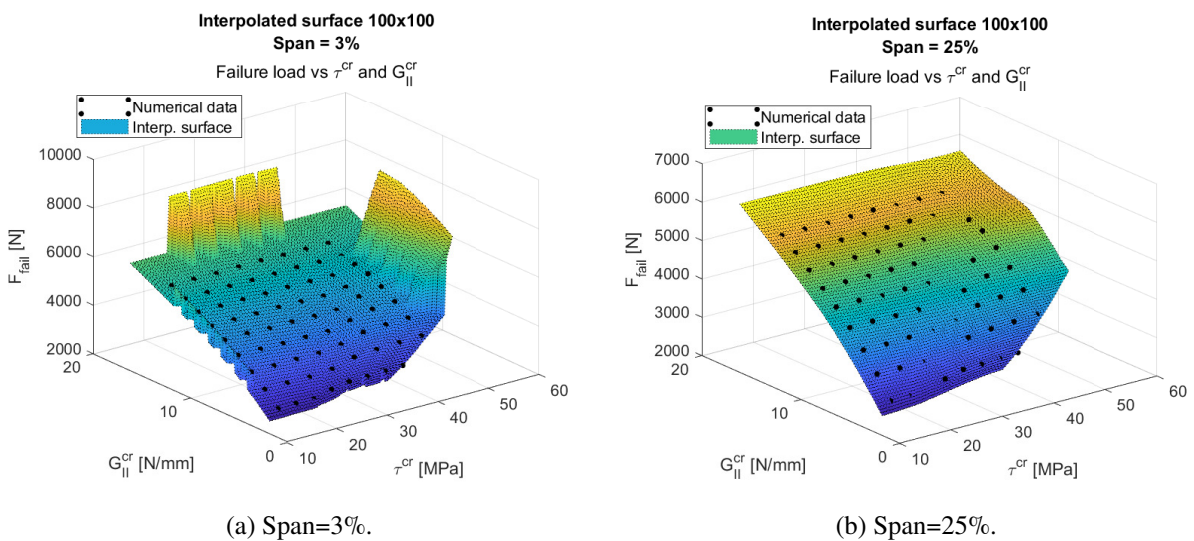


Figure 5.20 Surface interpolation with Curve Fitter MATLAB tool.

Figure 5.20 shows, for a surface of 100x100 points of discretization and an extended range of 20% for



each boundary of the values where the input points are calculated, the influence of the span parameter, which reflects what has just been explained. On the one hand, in Figure 5.20a, the numerical input data is very close to the interpolated surface with span of 3% (maximum error at points of 0.75 % ), whereas the surface far from the points is very distorted. On the other hand, Figure 5.20b represents a surface obtained with a span of 25% that differs more from the initial input (maximum error at points of 2.52%) but is much smoother, being valid farther from the  $G_{II}^{cr}$  and  $\tau_{cr}$  input domain.

In our case, this parameter will not have substantial effect because we will look for points inside the range of points, where the surface is smooth no matter the choice of the span. Therefore, it will be chosen small (3%) to be more accurate inside our range.

The input data, the type of interpolation, span and some additional options are inserted in the *fit* function of MATLAB. Code 5.1 shows a function as an example for creating the interpolated surface from x, y and z data.

**Code 5.1** MATLAB function to create an interpolated surface.

```
function [fitresult, gof] = createFit(xData, yData, zData,span)

    % Set up fittype and options.
    ft = fittype( 'lowess' );
    opts = fitoptions( 'Method', 'LowessFit' );
    opts.Normalize = 'on';
    opts.Span = span;

    % Fit model to data.
    [fitresult, gof] = fit( [xData, yData], zData, ft, opts );

end
```

In our case,  $xData = \tau_{vect}^{cr}$ ,  $yData = G_{II-vect}^{cr}$  and  $zData = F_{vect}^{fail}$ , which are the vectors showed in Equation 5.4. In addition, Code 5.2 shows how to call this function and how to interpolate any pair of  $G_{II}^{cr}$  and  $\tau_{cr}$  to obtain the estimated  $F^{fail}$ . It must be noticed that the blank vectors should be filled with our parameters and numerical results but they are empty to avoid a large and distracting code.

**Code 5.2** MATLAB script to call the interpolated surface.

```
%Define parameters and input data
span=0.03;
tau_to_fit = [];
GII_to_fit = [];
F_to_fit = [];

% Create the interpolated surface
[fitresult, gof] = createFit(tau_to_fit, GII_to_fit, F_to_fit,span);

% Calculate the failure loads by introducing the desired properties in
% matrix form
F_fitted=fitresult(tau_cr_mat,GII_cr_mat);
```

It is important to point out that  $\tau_{cr\_mat}$  and  $G_{II\_cr\_mat}$  are the desired input parameters in matrix form. These matrices must be same dimension and will result in a  $F_{fitted}$  matrix of same dimensions where each component is the failure load corresponding to that component properties. For example, if we desired to obtain the estimated failure loads from the original inputs (see Equation 5.4), these matrices would be 8x9, just like Equation 5.5 and Equation 5.6 show.

$$G_{II}^{cr-matrix} = \begin{bmatrix} 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \\ 3 & 4.375 & 5.75 & 7.125 & 8.5 & 9.875 & 11.25 & 12.625 & 14 \end{bmatrix} N/mm \quad (5.5)$$

$$\tau_{matrix}^{cr} = \begin{bmatrix} 14.00 & 14.00 & 14.00 & 14.00 & 14.00 & 14.00 & 14.00 & 14.00 & 14.00 \\ 17.53 & 18.43 & 18.43 & 18.43 & 18.43 & 18.43 & 18.43 & 18.43 & 18.43 \\ 21.05 & 22.86 & 22.86 & 22.86 & 22.86 & 22.86 & 22.86 & 22.86 & 22.86 \\ 24.58 & 27.29 & 27.29 & 27.29 & 27.29 & 27.29 & 27.29 & 27.29 & 27.29 \\ 28.11 & 31.71 & 31.71 & 31.71 & 31.71 & 31.71 & 31.71 & 31.71 & 31.71 \\ 31.64 & 36.14 & 36.14 & 36.14 & 36.14 & 36.14 & 36.14 & 36.14 & 36.14 \\ 35.16 & 40.57 & 40.57 & 40.57 & 40.57 & 40.57 & 40.57 & 40.57 & 40.57 \\ 38.69 & 45.00 & 45.00 & 45.00 & 45.00 & 45.00 & 45.00 & 45.00 & 45.00 \end{bmatrix} MPa \quad (5.6)$$

At this point, once the interpolation process is understood, we are ready to numerically solve the 3 overlap cases, calculate the interpolated surfaces and obtain the intersection curves between them and the test data which is compiled in Table 5.4.

The steps that are followed for each case to reach this objective are:

- Numerically solve the 72 cases by means of the loop codes (see Appendix A).
- Obtain an interpolated surface of 301x301 points. This dimension has been chosen because less points might not be enough to cut the planes without substantial error, whereas more points would significantly increase the computational time needed to make the interpolation (it currently takes around 10 seconds).
- Look for the intersection curve between the interpolated surfaces (which is composed of 90601 points) and the horizontal planes obtained from tests. To do that, the *find()* MATLAB function has been employed. The points that will trace the intersection curves are those contained in an interval extended 0.02% from the searched value. For example, in the intersection of the failure surface of the analysis with an overlap length of 5mm with the corresponding mean failure load from the test (4325N), the surface points will be selected to form the curve if they are in the range  $4325 \cdot [0.9998, 1.0002] = [4324.135, 4325.865]$ . A simplification of the code employed to obtain these curve points is shown in Code 5.3 to easily understand the process.

**Code 5.3** MATLAB script to find the intersection between horizontal planes and interpolated surface.

```
% Error of intersection range
Curv_err=0.0002;

% Failure data from experiments
%Test1      %Test2      %Test3      %Test4
R1_exp1=3650; R1_exp2=4000; R1_exp3=4400; R1_exp4=5250;

R1_exp_MEAN=(R1_exp1+R1_exp2+R1_exp3+R1_exp4)/4;
R1_exp_MIN=min([R1_exp1,R1_exp2,R1_exp3,R1_exp4]);
R1_exp_MAX=max([R1_exp1,R1_exp2,R1_exp3,R1_exp4]);
```

```

R1_exp_iter=[R1_exp_MIN;R1_exp_MEAN;R1_exp_MAX];

% Interpolated range: 301x301 points
tau_rang = []; %301x301
GII_rang = []; %301x301
F_rang=fitresult(tau_rang,GII_rang);

for k=1:3 %min, mean, max

    % Find the position of the curve points
    [Row,Col]=find(F_rang>=R1_exp_iter(k)*(1-Curv_err) & F_rang<=R1_exp_iter(k)*(1+
        Curv_err));
    Pos=[Row,Col];

    % Find the properties for those chosen points
    tau_curv=[];      GII_curv=[];
    for i=1:size(Pos,1)
        Pos1=Pos(i,1);
        Pos2=Pos(i,2);
        tau_curv=[tau_curv;tau_rang(Pos1, Pos2)];
        GII_curv=[GII_curv;GII_rang(Pos1, Pos2)];
    end

    % Arrange
    [tau_curv_ord,Indx]=sort(tau_curv);
    GII_curv_ord=GII_curv(Indx);

    % Safe curve properties (tau_cr and GII_cr)
    if k==1
        tau_curv_5mm_MIN=tau_curv_ord;
        GII_curv_5mm_MIN=GII_curv_ord;
        save('05b_Curves_Data\a_Curves_5mm_MIN.mat','tau_curv_5mm_MIN','
            GII_curv_5mm_MIN')
    elseif k==2
        tau_curv_5mm_MEAN=tau_curv_ord;
        GII_curv_5mm_MEAN=GII_curv_ord;
        save('05b_Curves_Data\a_Curves_5mm_MEAN.mat','tau_curv_5mm_MEAN','
            GII_curv_5mm_MEAN')
    elseif k==3
        tau_curv_5mm_MAX=tau_curv_ord;
        GII_curv_5mm_MAX=GII_curv_ord;
        save('05b_Curves_Data\a_Curves_5mm_MAX.mat','tau_curv_5mm_MAX','
            GII_curv_5mm_MAX')
    end
end
end

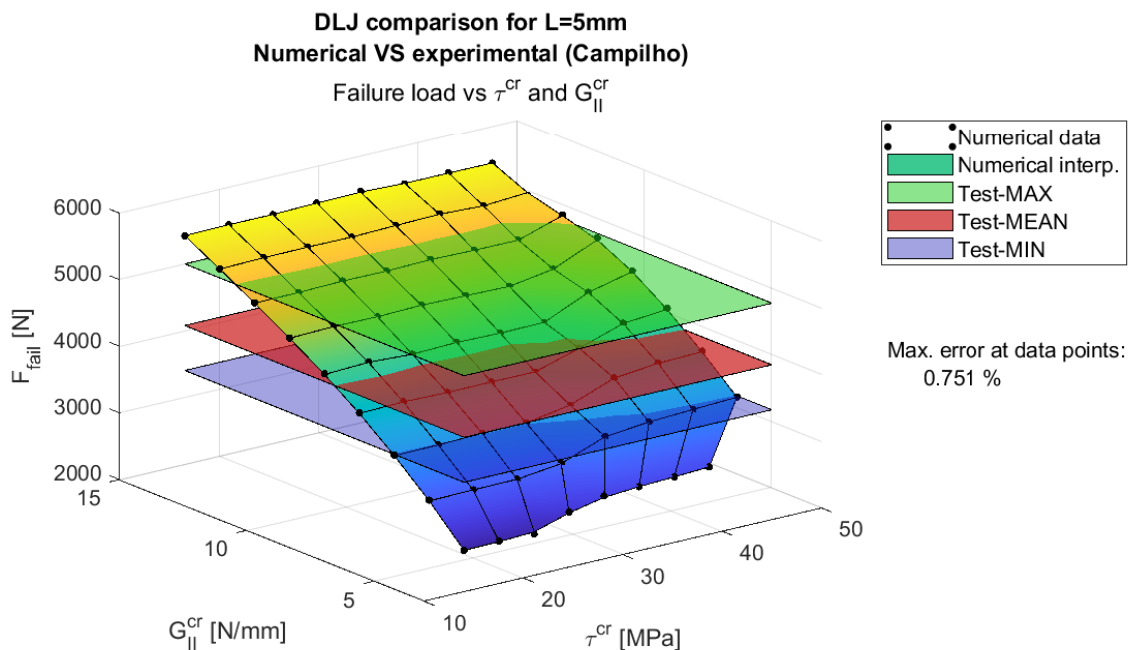
```

Once the process that have been carried out is clear, the final results concerning surfaces and curves will be exposed for the three overlap lengths.

## Surfaces

Firstly, the numerical interpolated surfaces and the horizontal planes are represented in Figure 5.21, Figure 5.22 and Figure 5.23 for overlap lengths of 5, 10 and 20mm, respectively.

- The black points represent the numerical data obtained from the PMTE-SC formulation for each of the pairs of  $G_{II}^{cr}$  and  $\tau_{cr}$ .
- The assorted coloured surface is the interpolated surface from the numerical results (although the legend shows dark green colour).
- The horizontal single-coloured transparent planes correspond to the failure loads of the experiments from [3]. From top to bottom, they represent the maximum (lime green), the mean (red) and the minimum (blue) value. Therefore, the surface of interest will be contained within this range.
- The error showed in these pictures indicates the maximum error between the interpolated surface and the numerical results.



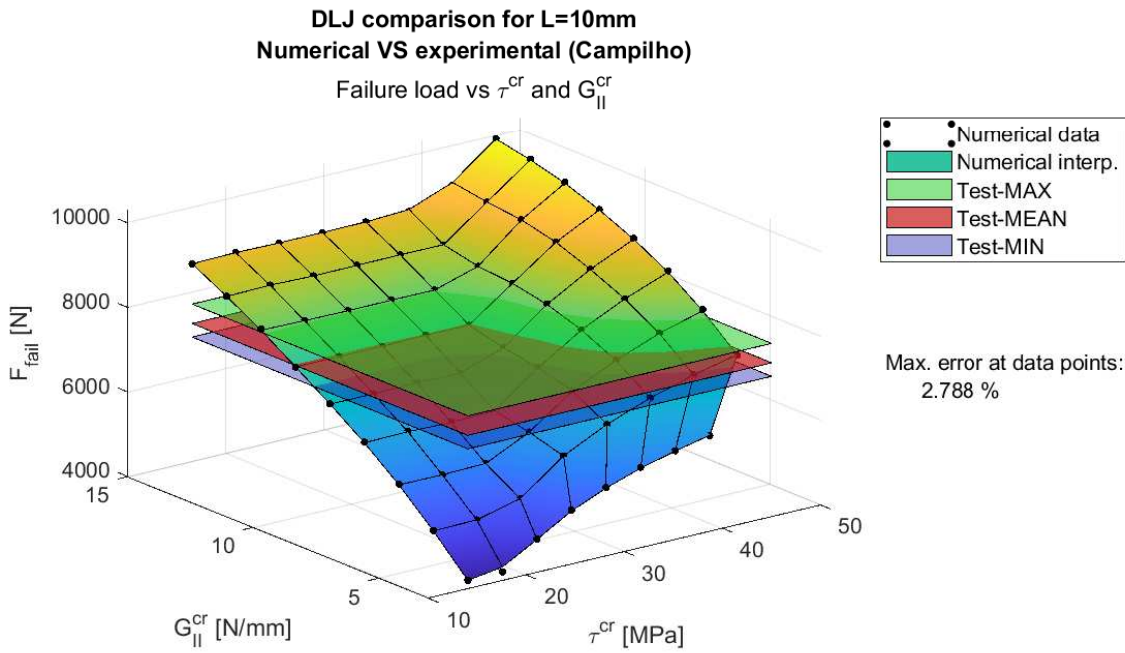
**Figure 5.21** L=5mm - Failure surface: numerical vs experimental (min., mean and max.).

In Figure 5.21 we can see an homogeneous slope for the failure surface, which will involve almost horizontal intersection curves. Furthermore, the horizontal planes are quite separated between each other, which is caused due to the substantial differences in the test results for the overlap length of 5mm. This fact will imply a wide area (Figure 5.28a) where adhesive properties will be searched.

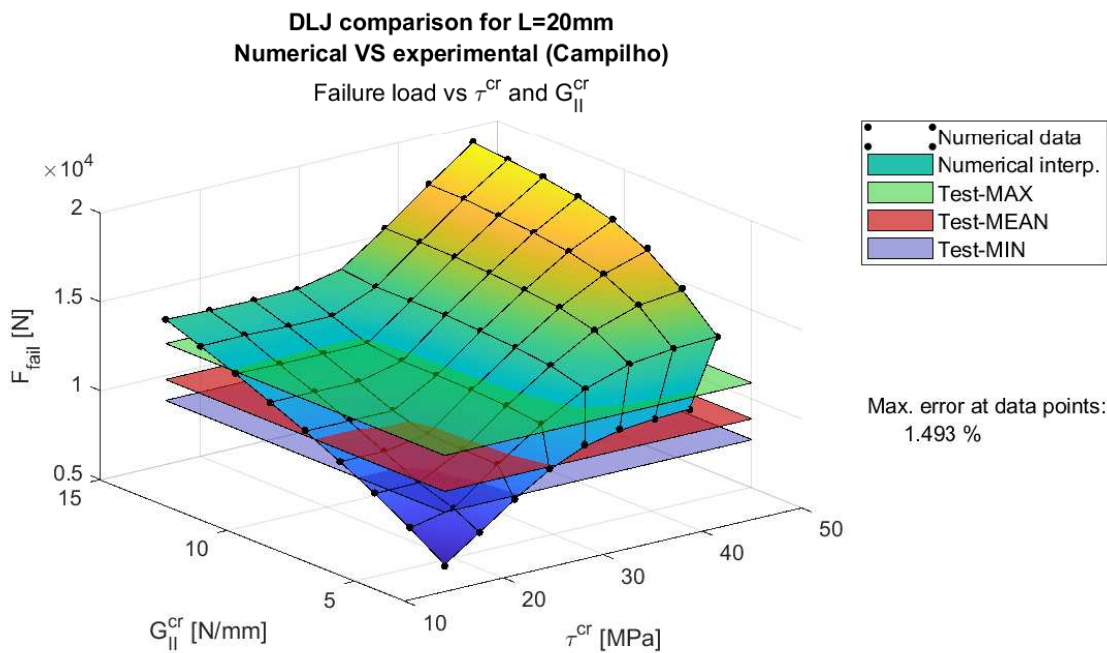
In opposition to tests with overlap length of 5mm, those from 10mm seem more precise since the horizontal planes are much closer (see Figure 5.22). This will imply small area (Figure 5.28b) for the determination of the adhesive parameters. In addition, we see that the failure loads depends more on  $\tau_{cr}$  than for overlap length of 5mm.

Finally, Figure 5.23 shows the numerical failure surface and horizontal planes from tests for overlap length of 20mm. In this case, an intermediate area (Figure 5.28c) can be guessed with respect to previous overlap lengths.

Furthermore, in both overlap lengths of 10 and 20mm, a change in the surface behaviour is observed. Depending on which parameter dominates, the slopes change. For example, similar values of failure forces are obtained for fixed  $G_{II}^{cr}$  and small values of  $\tau_{cr}$ ; or for fixed values of  $\tau_{cr}$  and large  $G_{II}^{cr}$ . Meanwhile,



**Figure 5.22** L=10mm - Failure surface: numerical vs experimental (min., mean and max.).



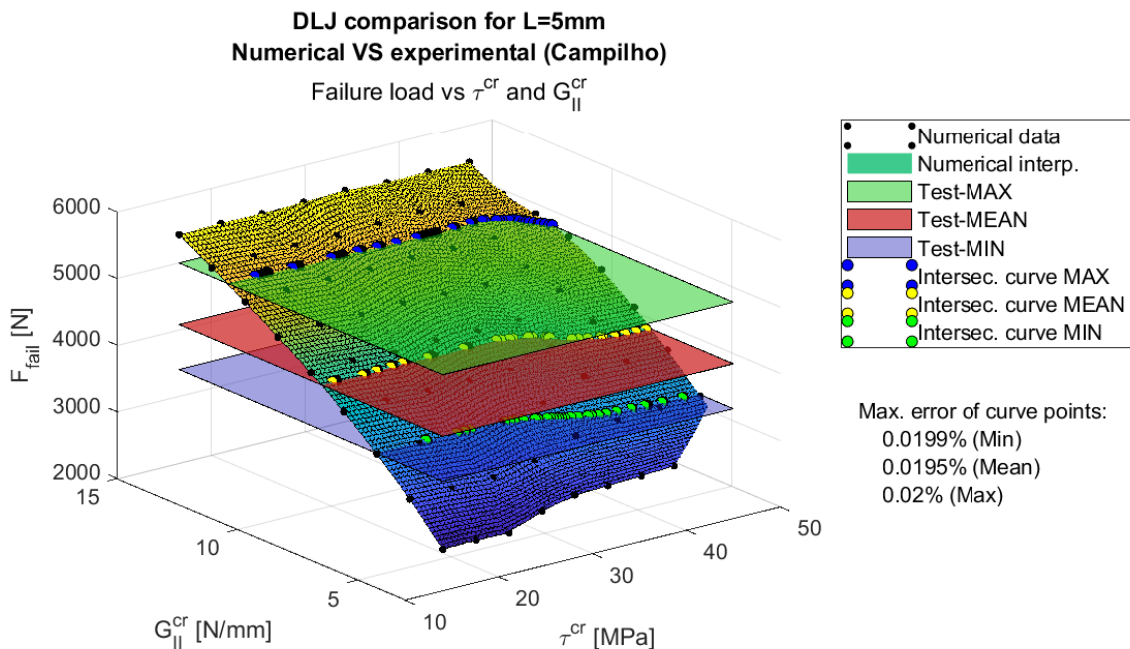
**Figure 5.23** L=20mm - Failure surface: numerical vs experimental (min., mean and max.).

for overlap length of L=5mm,  $\tau_{cr}$  has little impact, since for fixed  $G_{II}^{cr}$ , the failure load does not vary substantially.

### Intersection curves

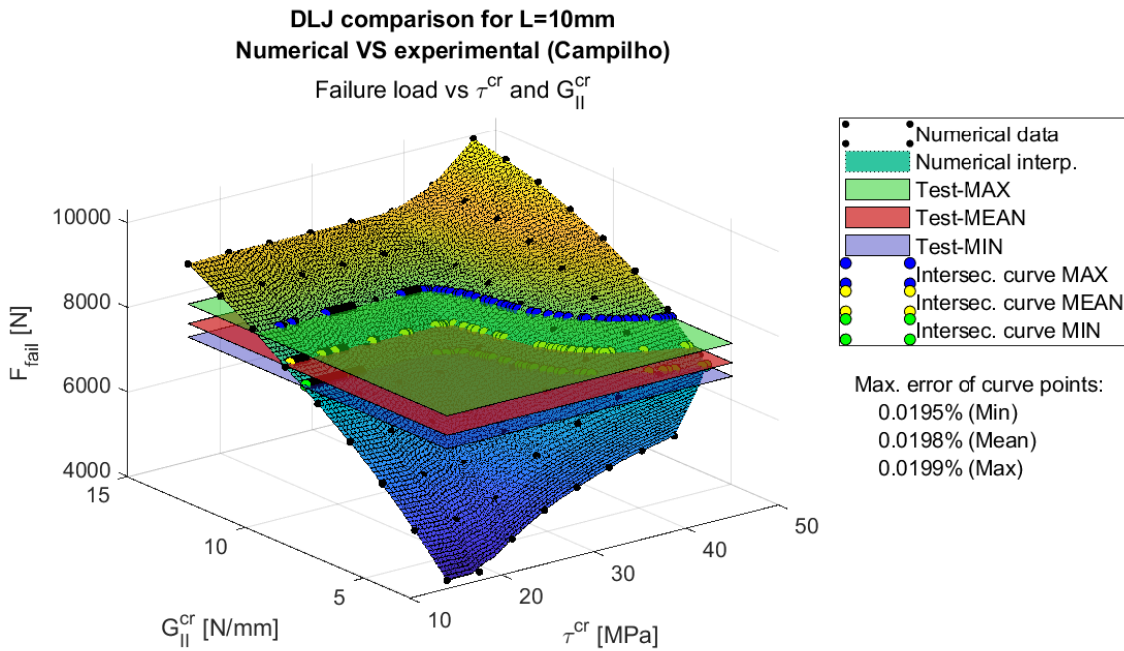
The intersection curves between the failure surfaces and horizontal planes are shown in Figure 5.24, Figure 5.25 and Figure 5.26 for overlap lengths of 5, 10 and 20mm, respectively.

- The surface is discretized to be able to caught the intersection with the horizontal planes. This discretization consist of the small hundreds of squares and they must not be confounded with the numerical data (black points).
- The coloured circles represent the points of the discretized interpolated surface that intersect with the horizontal plane under the criteria of the previously mentioned 0.02% failure load range. In particular, blue points represent the intersection curve with the maximum failure load of the tests, yellow for the mean and green for the minimum one.
- The error shown in the figures correspond to the maximum error of the points with respect to maximum, mean and minimum load planes from experiments. They must be equal or smaller than 0.02%, which was the chosen criterion.

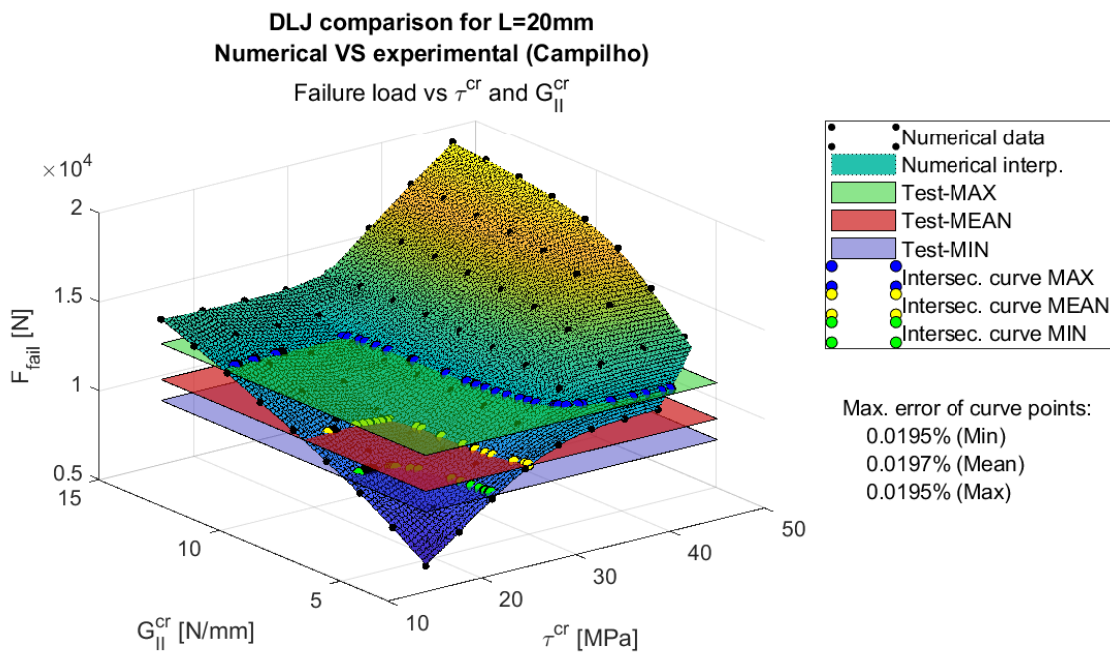


**Figure 5.24** L=5mm - Failure surface and intersection curves.





**Figure 5.25** L=10mm - Failure surface and intersection curves.



**Figure 5.26** L=20mm - Failure surface and intersection curves.

Once the numerical interpolated surfaces have been obtained and the points which form the intersection curve are determined, we can represent them altogether to see the correlation between them.

Firstly, in Figure 5.27, the intersection curves ( $G_{II}^{cr}$  vs  $\tau_{cr}$ ) for the mean failure loads of the tests are represented for every overlap length: 5mm in red, 10mm in blue and 20mm in green. As it typically occurs, they do not converge into a certain point due to the imperfections of the tests, scattering, etc.

Therefore, it is appropriate to use the whole set of tests and represent the areas between maximum and minimum loads, which correspond to the admissible values for  $\tau_{cr}$  and  $G_{II}^{cr}$  for every test. In Figure 5.28,

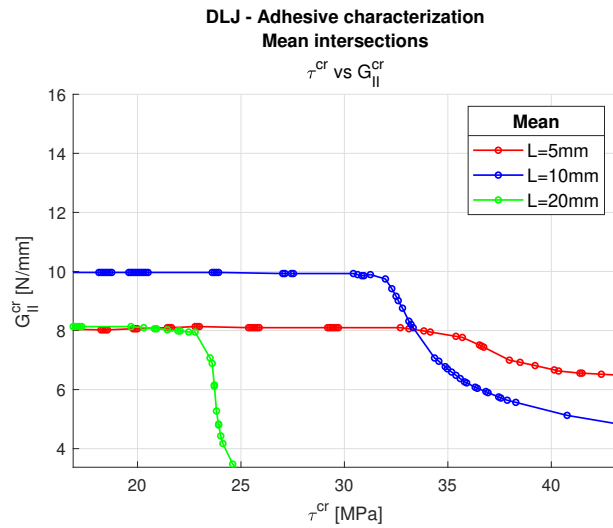


Figure 5.27 Intersection curves at mean values of experiments for 3 overlap lengths.

these admissible areas are shown for overlap lengths of 5, 10 and 20mm.

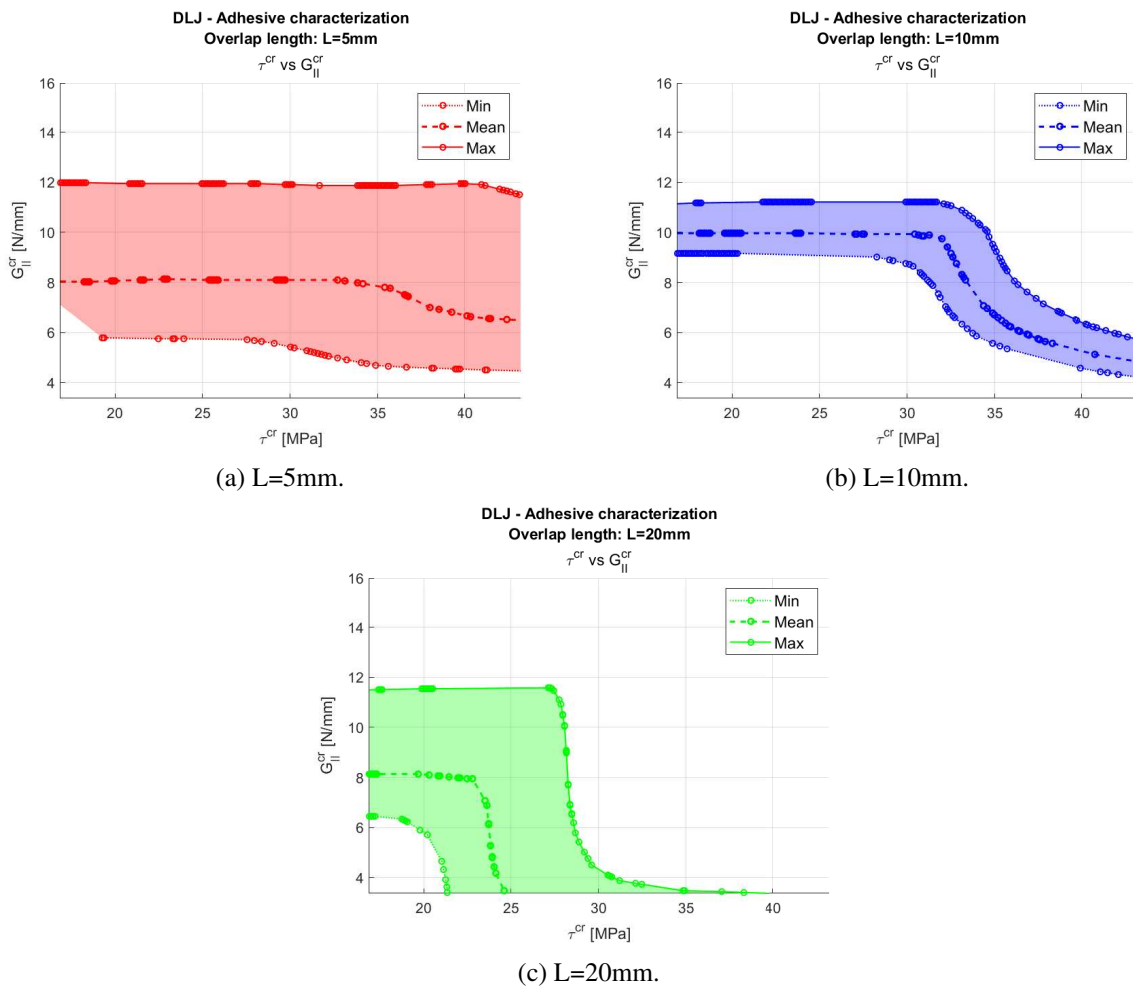


Figure 5.28 Intersection curves at min., mean and max. values of experiments for 3 overlap lengths.

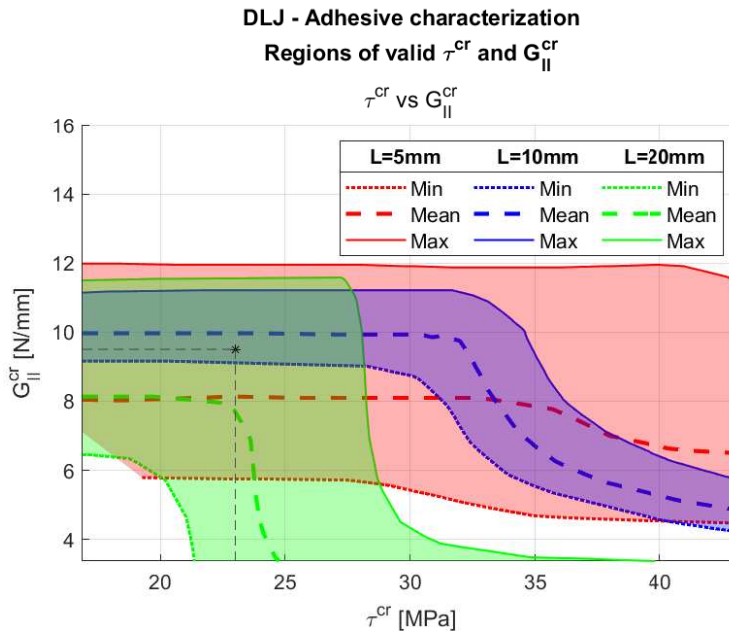
From these figures (and those representing the surfaces), it is appreciated that the failure load is more



dependent on the fracture toughness ( $G_{II}^{cr}$ ) because the curves are close to be horizontal. This happens because the energetic criterion is harder to be satisfied. For example, if for a fixed value of  $G_{II}^{cr}$ , we analyse two cases with  $\tau_{cr,1} < \tau_{cr,2}$ , the second would require higher loads to fulfill the stress criterion; however, it might not be enough to satisfy the energetic criterion for none of them. Therefore, the failure load will be the same for both even-though the second case has a larger  $\tau_{cr}$ .

This effect especially appears for shortest overlap length. Meanwhile, for the largest overlap length, this only appears for small values of  $\tau_{cr}$ . This might be caused because for small overlap lengths, shear stresses in adhesive are flatter, just like it was seen in Figure 5.8a, so the energetic criterion dominates over the stress criterion since there are no stress concentrators. On the contrary, for larger overlap lengths, this happens for small  $\tau_{cr}$ , but for the higher ones, the stress criterion is later fulfilled (specially for low  $G_{II}^{cr}$ ) and the curves tend to be vertical (see the green curves tendency at  $G_{II}^{cr} = 5$ ).

If we represent these areas altogether (Figure 5.29), we can see the overlap between them, being able to determine a pair of values for our parameters that properly satisfy the 3 regions and are closer to the mean result of the experiments, rather than those from maximum or minimum. As the figure shows, the chosen combination for critical shear stress and mode II fracture toughness is  $\tau_{cr} = 23\text{MPa}$  and  $G_{II}^{cr} = 9.5\text{N/mm}$ , respectively, which together with the shear stiffness of  $k_t = 308\text{MPa/mm}$ , close the characterization of the adhesive LEBIM by means of reverse engineering and experiments from the literature.



**Figure 5.29** Valid regions of  $\tau_{cr}$  and  $G_{II}^{cr}$  for 3 overlap lengths.

With these results in mind, if we obtain the relation between critical and maximum tangential stresses, we obtain that  $\frac{\tau_{cr}}{\tau_{max}} = \frac{\tau_{cr}}{\sqrt{2 \cdot G_{II}^{cr} \cdot k_t}} = 0.3$ . Then, we observe that our first estimation of 0.35 was quite close. Therefore, this kind of estimation can be taken into consideration to have a first approach of  $\tau_{cr}$  in case we know  $G_{II}^{cr}$  (or can be estimated).

Finally, in order to check the adjustment of the parameters, analyses for every overlap length with the chosen parameters were launched. Figure 5.30 shows the resulting load vs displacement curve (in red) for the overlap length of 5mm under the PMTE-SC methodology, as well as the already shown experiments and numerical results from Campilho. As it can be seen, our result is contained in the range of the experiments and it has the same slope than the mean of them because the stiffness  $k_t$  was obtained from experiments with this overlap length.

In addition, as summary, results for every overlap length are shown in Figure 5.31, demonstrating that

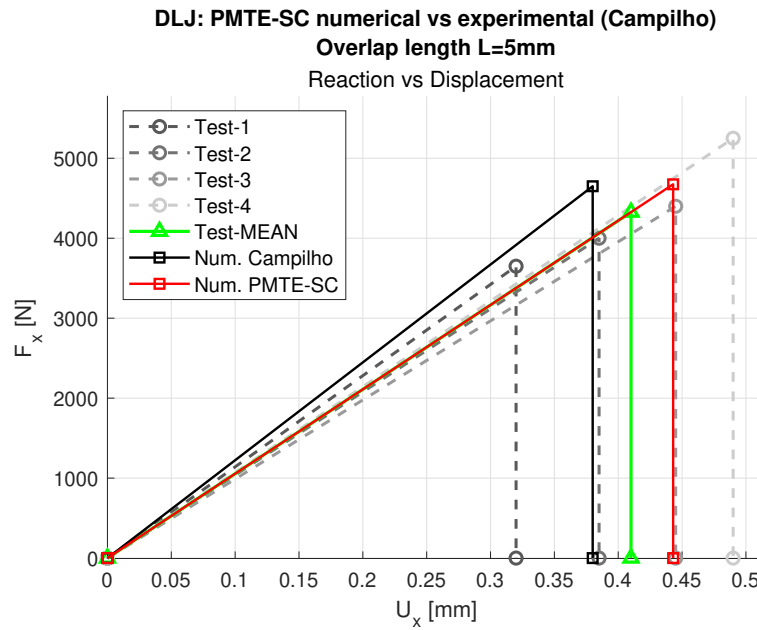


Figure 5.30 PMTE-SC numerical results vs experiments. Reaction with respect to displacement for an overlap length of 5mm.

our numerical results with the adjusted adhesive parameters under PMTE-SC (represented with circles) are contained in the range of the experiments from the literature.

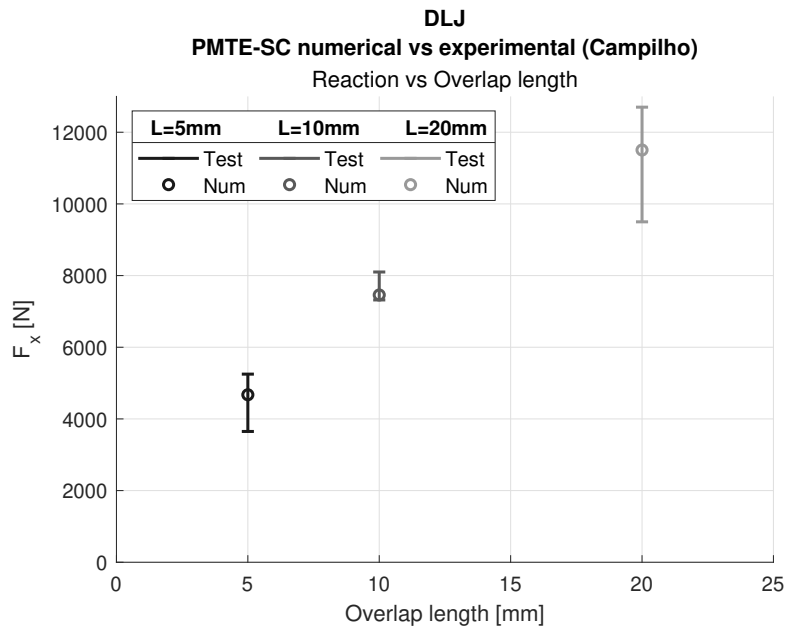


Figure 5.31 PMTE-SC numerical results vs experiments. Failure load for every overlap length.

### 5.3.3 Curves of energetic behaviour

In this final section, the total energy curves with respect to crack length for DLJ models will be plotted. The numerical curves have been obtained by means of the tool developed in this project with the properties of the adherends and adhesive that have been obtained in the just developed subsection 5.3.2. The curves are obtained under both load control (Figure 5.32) and displacement control (Figure 5.35) for the already studied case of adhesive with 10mm length.

Just like it was previously done in section 4.6 for DCB, the nomenclature of the markers and line shapes is the following:

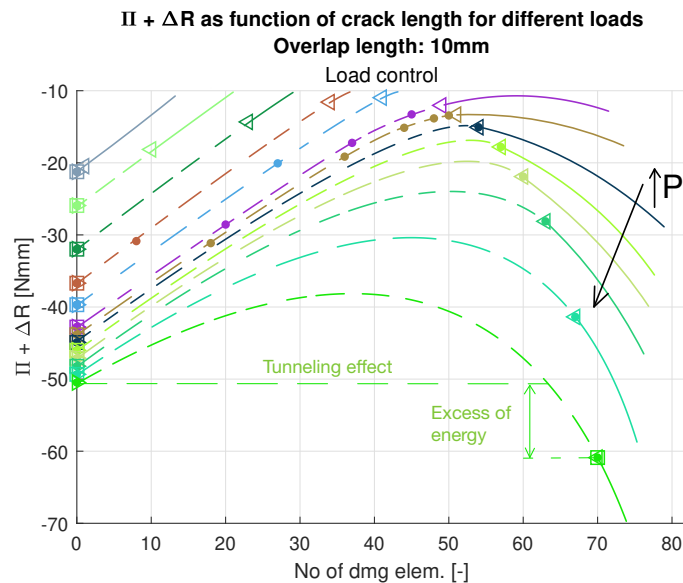
- Triangles pointing to the right ( $\triangleright$ ) indicate the lower limit of the interval that represents the elements that satisfy the stress criterion. This limit corresponds to the first start of the minimization ( $n=1$ ). Since for the current case there is no previous damage (the fracture might involve the whole adhesive length), they are all positioned at the y-axis.
- Triangles pointing to the left ( $\triangleleft$ ) indicate the upper limit of the just mentioned interval, which corresponds to the second start of the minimization ( $n=2$ ).
- Dashed lines represent the curve portion where elements satisfy the stress criterion.
- Circular filled markers ( $\bullet$ ) represent intermediate points of the minimization process. Those points where there is both triangle and circular markers imply they are local minimums.
- Quadrangular markers ( $\square$ ) represent the final point of the minimization, which indicates the number of elements that satisfy both stress and energetic criteria and the minimum total energy, which is associated to the final state.

Figure 5.32 and Figure 5.35 show the just mentioned curves. It must be noticed that both represent the total energy with respect to crack length (represented by the number of finite elements, taking into account that the adhesive contains 80 elements of 0.125mm). However, this energy is referred as potential plus released for load control, and internal plus released for displacement control. This is because the external work is null for the last case so  $\{\Pi = U\}_{DC}$ . Also, it must be said that the load under displacement control is a translation represented with  $U$  to stand out the ascending load direction, but it should not get confused with the internal energy. Furthermore, it must be reminded that the numerical information given by the tool is the different points: triangles, circles and squares, which are located by means of the energy and the number of finite elements associated to the minimization process. However, the curves are manually drawn by following the information the numerical tool provides us, the read literature and the knowledge obtain along the project. Therefore, they should only be taken as reference for educative purposes. In fact, the lack of information for displacement control curves after the triangles pointing to the left involves that the continuous part of the curve have been omitted. Finally, it must be pointed out that due to the small load increment values that were used to run the analysis, many curves could have been obtained but the graph would have been confusing. Therefore, only some of them have been plotted to be able to understand the different behaviours, but the load increments between them are not constant.

By comparing both graphs, the first thing we see is every curve is concave in disagreement with curves for DCB tests under displacement control which are convex (see subsection 4.6.1). This gives us the clue that the cracks will be finite and the propagation will be unstable for both controls.

### Load control

Firstly, we will make some comments for the load control graph (Figure 5.32).



**Figure 5.32** Numerical curves of energetic behaviour for overlap length of 10mm under load control.

Under small loads (upper curves) the energy is not enough to create cracks. As it is increased, the stress criterion (discontinuous lines) are larger. For the first 7th curves, both minimizations ( $n=1$  and  $n=2$ ) end in 0 damaged elements (the square is at the left) because the discontinues curve is only ascending. For the dark curve, we can see the right-hand triangle with a circle for the first time, which means that the minimization with  $n=2$  arrives to a local minimum which would induce a crack growth, but the absolute minimum is reached for  $n=1$  and the crack does not appear. For higher loads, the curves are more concave and the descending part is more relevant, up to the final lime green curve in which the total energy is smaller for the minimization of  $n=2$ . It can be seen that there is an excess of energy so the crack would actually appear for an intermediate case between the last two curves. In this case, the load step between them is 1N so the loads for the final two curves are 87.236N and 88.236N, respectively, and the real load which provokes a crack growth will be between them. As it can be seen in the last curve, the number of element that satisfy the stress criterion is 70 (out of the total 80 elements that compose the adhesive). This means that, as Figure 5.33 shows, there is still undamaged adhesive after the crack growth. However, if we allow the tool to make another iteration for step  $m$ , which consist of analyse the specimen under the same load but with this initial crack, the crack will progress and the adhesive will completely break. This behaviour is very common under load controls.

Figure 5.33 represents the damage variable of the UMAT (SDV1), which is 1 for undamaged elements and 0 for the broken ones, for the specimen we are analysing after the crack growth ( $k=32$ ,  $m=1$ ,  $n=2$ ,  $j=2$ ) at the adhesive. A zoom focused on the adhesive is make with a deformation scale factor of 3 to be able to perceive the discretization and damage contour.

In addition, with respect to stresses, what is expected and was already mentioned in section 5.2 is now obtained. In Figure 5.34, stresses at the adhesive for the instant prior to crack onset are plotted.

It must be mentioned that normal stresses are those in the vertical direction, which is direction 1 at the definition of the elements by LEBIM so we have represented  $S_{11}$ , but it is the global  $Y$  direction. As we expected, the adhesive is under tension at the right (24.66MPa) and compression at the left (21.63MPa). In fact, we see the already mentioned anti-symmetry of the normal stresses but higher peaks at the right side of the adhesive due to the numerical boundary conditions.

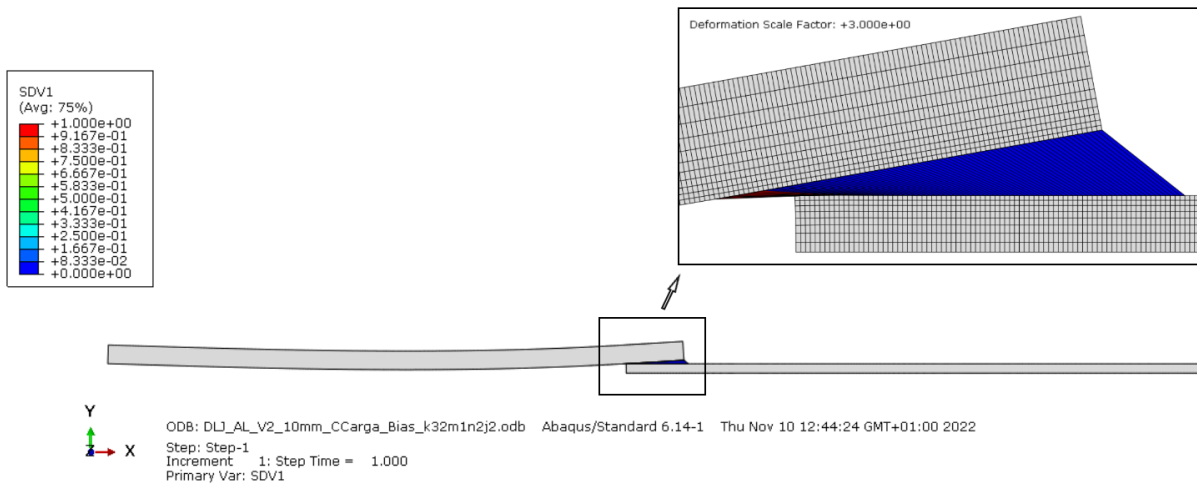
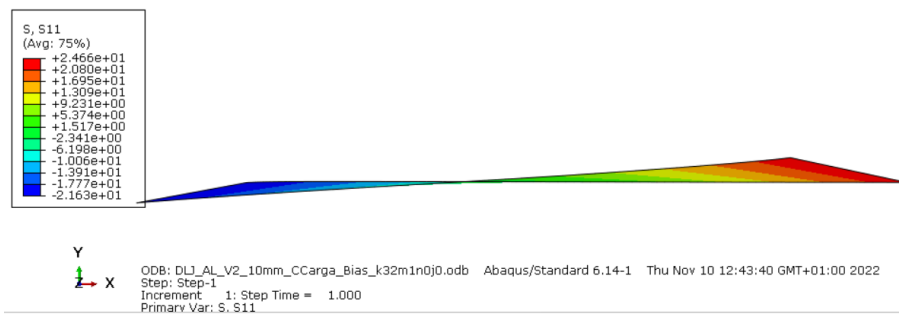
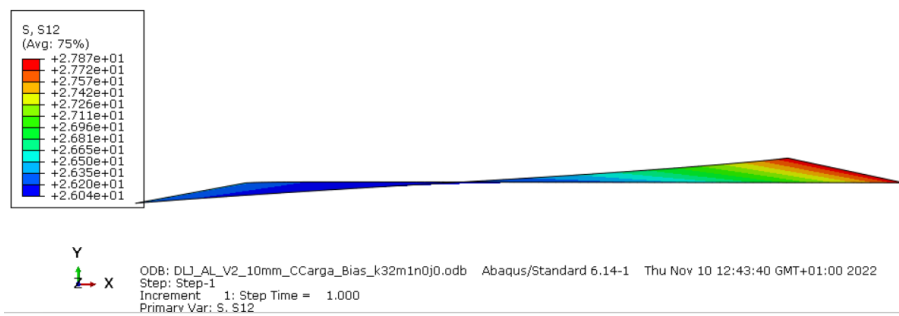


Figure 5.33 Numerical breakage of adhesive under load control.

With respect to shear stresses ( $S_{12}$ ), we see an homogeneous distribution, just like we saw in figures from section 5.2, from 26.04MPa at the left to 27.87MPa at the right. It must also be pointed out that normal and shear stresses are of the same magnitude in this kind of test so none of them should be neglected.



(a) Normal stresses ( $S_{11}$ ).

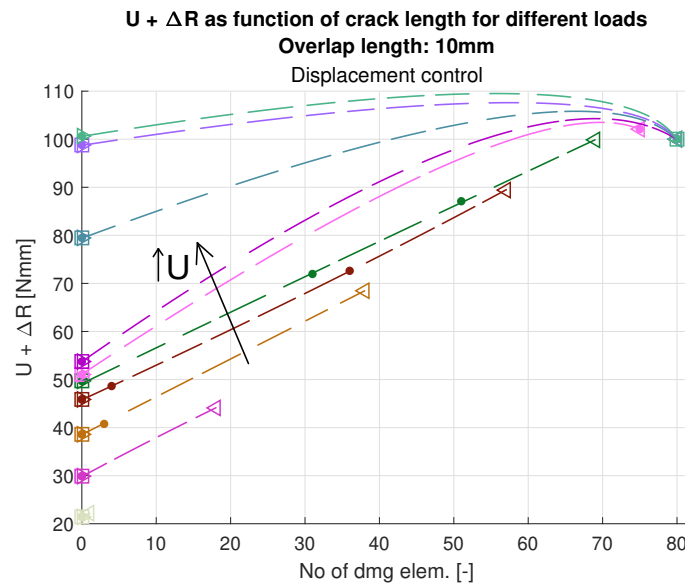


(b) Shear stresses ( $S_{12}$ ).

Figure 5.34 Stresses at adhesive.

### Displacement control

Secondly, the curves under displacement control (Figure 5.35) will be commented.



**Figure 5.35** Numerical curves of energetic behaviour for overlap length of 10mm under displacement control.

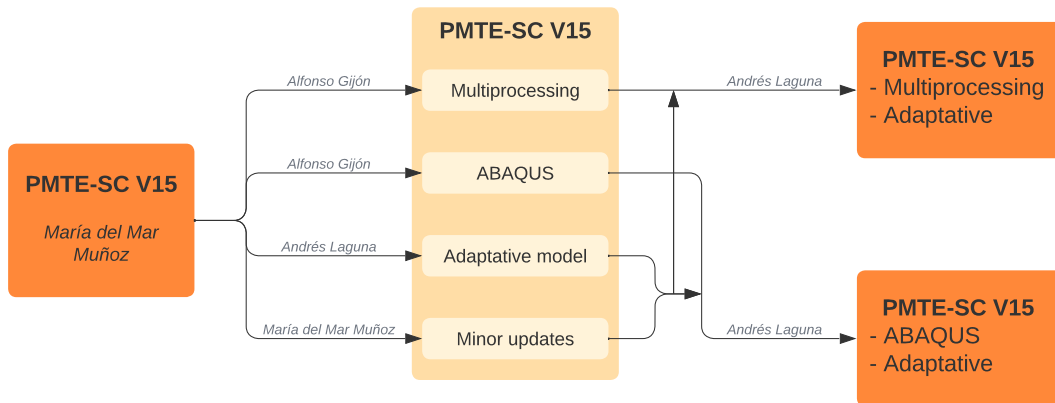
In this case, the load (displacement) increases from bottom to top curves. Similarly to what happened under load control, for the first 5 curves (until the green one), although the stress criterion is fulfilled, the ascending behaviour of the total energy curve does not involve crack growth (the square points are at left beginning of the curves). For the next 4 curves, with higher loads, a descending part of the curve provokes that the minimization for  $n=2$  suggest a finite crack growth but the absolute minimum is reached at  $n=1$  so it does not happen. For the upper curve (green) it is the first time the right-hand triangle has less energy than the left-hand triangle. This way, the total energy between initial and final states decreases and the crack growth takes place. In particular, every element of the adhesive (80 in total) fulfills the stress criterion so the finite crack will propagate across the whole adhesive instantaneously.

# 6 Unification of codes

*A goal is a dream with a deadline*

NAPOLEON HILL

During this project and last years, the *PMTE-SC* codes have been developed by several people at the same time, leading to codes with different procedures and functionalities but sharing a goal. For this reason, with the objective of not losing relevant work and advances, in this section, we will try to compile everything in two blocks of codes. This progress has been represented in Figure 6.1 by means of a flowchart, which also includes the people involved in each development.



**Figure 6.1** Evolution of PMTE-SC codes.

The first block, which is referred as multi-processing, was developed in order to solve the N steps simultaneously. Currently, we only solve two steps of N, where all the elements which fulfill the stress criterion are initially damaged or without damaged, respectively. However, it is expected to soon implement intermediate cases where those elements are partially damaged. At that moment, the paralelization will be crucial to reduce computational costs.

The second block is a code that is directly launched from ABAQUS (run script), instead of Python. The purpose of this procedure is to be able to run dense meshed problems simultaneously in the CLUSTER. This is possible because the code does not update the *inp* file and import the model like the multiprocessing code does, but it loads the *inp* file from ABAQUS, and loads and boundary conditions are set afterwards.

More details about the codes for both ways of launching and how to prepare the scripts and files for new problems can be found in Appendix C.





# 7 Conclusions

---

*The world does not care how young or weak you are, step up or die*

ANONYMOUS

This project has been mainly focused on the prediction of crack onset and propagation for different adhesive joints by means of the Principle of Minimum Total Energy subjected to Stress Condition (PMTE-SC) developed by the Department of Elasticity and Resistance of Materials of the University of Seville (US).

- Firstly, a global review about the topic was made, including Linear Elastic Fracture Mechanics (LEFM), Linear Elastic Brittle Interface Model (LEBIM), Finite Fracture Mechanics (FFM), Coupled Criterion of Finite Fracture Mechanics (CCFFM), as well as the formulation of the Principle of Minimum Total Energy subjected to Stress Criterion (PMTE-SC). This has been very useful to refresh main concepts about Fracture Mechanics to be able to face the rest of the project.
- Then, the numerical implementation of the tool has been explained, including the nomenclature employed, a description and explanation about the minimization process and some examples from the literature with analytical curves to better understand the algorithm.
- Thanks to the PMTE-SC formulation we have been able to analyse two joints. Some comments and results were firstly shown for the DCB, which fails under mode I of fracture. However, the Elasticity Group has previously worked a lot around this type of joint. For this reason, it is the DLJ where more emphasis was placed. This joint breaks under mixed fracture modes (I and II), demonstrating that the PMTE-SC formulation is able to predict mixed failure modes satisfactorily.
- Furthermore, reverse engineering has been successfully employed to be able to characterize the adhesive LEBIM properties ( $k_t$ ,  $\tau_{cr}$  and  $G_{II}^{cr}$ ) by means of experiments from the literature.
- In addition, existing codes have been understood and further developed to achieve advanced functionalities and allow two ways of launching: from ABAQUS and Python.
- Lastly, a way to run python scripts by means of an executable file (.exe) has been developed in Appendix B. This executable file allows to compress the main script and secondary files that are called. This is extremely useful to run Python scripts without showing them, being possible to share the functionality without revealing the formulation.

## 7.1 Future Work

Finally, some lines of research for future works are proposed:

- More joints can be studied to keep testing the effectiveness of the tool that has been developed during last years. One possibility is the Single Lap Joint (SLJ), which can be modelled like the DLJ of this project removing the symmetry condition. Although it might be initially thought that this joint is very similar to the DLJ, it must be taken into account that the analysis would not be linear anymore because the absence of symmetry induces an eccentricity in the load that provokes a moment. This moment depends on the eccentricity, but this eccentricity also depends on the moment, appearing a coupling between them. Therefore, the problem should be solved in several steps, in opposition to the analyses solved in this project, which were solved in one single step.
- With respect to the materials, no appropriate DLJ experiments with composite adherends were found in the literature. However, it is proposed to perform those tests and replicate the model which has been developed in this project to see if it still works with composite materials. In fact, it would be very useful because CFRP-Aluminium joints are highly common in the aeronautic industry.
- Although the codes are prepared to make the energetic minimization from multiple damage initiations (more than 2), as previously mentioned, it is currently being done with only two:  $n=1$  (no damage) and  $n=2$  (damage of every element satisfying stress criterion). Then, it is proposed to formulate new initiation criteria to be able to caught local minimums in more complex energetic curves that would not be perceived with current initiations, although those 'complex curves' have not appeared yet.

# Appendix A

## Scripts to launch parameter loops

---

In order to be able to sweep in two parameters and run the PMTE-SC codes many times by only one launch, a script was developed, see Code A.1.

**Code A.1** PMTESCPython\_launcher.py script to launch parametric loops.

```
#
#
#
# MAIN con bucle que llama al main original. Version con matriz de tau_cr
# para evitar incompatibilidades
import math
from os import path, mkdir, getcwd
from subprocess import call
from shutil import rmtree
import PMTESC
import time
import numpy as np
import numpy

start = time.time()

##### INPUTS -->
# SI SE PONE N_tau_cr=1 & N_GII_cr=1, se puede correr un unico caso
# SI el INP no tiene la cadena de la vble Adhesive_prop_previo,
# se lanza con esas propiedades como si no existiera este script
kt=308.0
kt_kn=0.2308 #k/kn
lambdaHS=0.5
h_adh=0.01 #adhesive thickness
kn=kt/kt_kn

N_tau_cr=8 #numero de tau_cr que barremos. Si solo queremos 1, coge el
tau_cr_min
tau_cr_min=14 #min tau_cr para barrer
tau_cr_max=45 #max tau_cr para barrer. Aunque luego se recalcula para que
sea compatible (puede ser mas pequeno)

N_GII_cr=9 #numero de GII_cr que barremos. Si solo queremos 1, coge el
GII_cr_min
```

```

GII_cr_min=3
GII_cr_max=14

# Propiedades del adhesivo iniciales. Si queremos las propiedades de este
# scrip, ponemos la siguiente linea en el .inp, tal cual esta aqui.
# Hace un cambio de linea por lo que si hay espacios distintos, no lo
# cogera bien
# De la misma forma, si hay otras propiedades en el inp, usara esas y se
# puede usar este script para lanzarlo ignorando las propiedades que se
# calculan.
Adhesive_prop_previo='10.0, 1.0, 0.5, 0.25, 0.01, 1.0'

# Nombre del main original que se llama en cada iteracion
Main_called = 'PMTESCpython_main_called.py'

# <--INPUTS #####

#vector de tau y GII en funcion de los valores que hemos elegido antes
tau_cr_list=list(np.linspace(tau_cr_min, tau_cr_max, N_tau_cr))
GII_cr_array=np.linspace(GII_cr_min, GII_cr_max, N_GII_cr)
GII_cr_list=list(GII_cr_array)

#vector de tau maximo compatible con GII_cr
tau_max_vec=0.9*numpy.sqrt(2*kt*GII_cr_array)
tau_cr_mat=[]

#Reformulacion de tau_cr para que sea compatible. Hace un linspace entre
# el minimo y el maximo admisible (que es el MINIMO entre el definido al
# ppio [tau_cr_max] o el calculado en [tau_max_vec])
for i in range(len(GII_cr_list)):
    tau_cr_mat.append(list(np.linspace(tau_cr_min,min(tau_cr_max,tau_max_vec[i]),
    N_tau_cr)))

# Borrar y crear directorio de Salida.
# En esta carpeta se genera desde el PMTESCmain_called.py una carpeta por
# cada pareja de tau y G que tengamos.
# Es decir, habra [N_tau_cr x N_GII_cr] carpetas
# Si N_tau_cr=2 y N_GII_cr=3, 6 carpetas llamadas:
# FFM_optimizada_1_1, FFM_optimizada_1_2
# FFM_optimizada_2_1, FFM_optimizada_2_2
# FFM_optimizada_3_1, FFM_optimizada_3_2
# Esto se gestiona mediante indices: index1 (para G) e index2 (para
# tau)
actual_directory = getcwd()
working_directory=actual_directory
working_directory_salidas=working_directory+'/Salidas'
if (path.exists(working_directory_salidas) == True):
    rmtree(working_directory_salidas)

mkdir(working_directory_salidas)

# Leer INPUT para coger el nombre del INP

```

```

fileINPUT = open('inputFFMLEBIM.txt','r')
flag = fileINPUT.readline()
name_INP = fileINPUT.readline().rstrip() + '.inp'

#Barrido en G y tau
index1=0
for GII_cr in GII_cr_list:
    print('GII_c='+str(GII_cr))
    index1=index1+1
    index2=0
    for tau_cr in tau_cr_mat[index1-1]: # -1 para empezar en 0
        print('\ttau_cr='+str(tau_cr))
        index2=index2+1

        # calculo de prop dependientes de GII_cr y tau_cr
        GI_cr=GII_cr*(math.sin(lambdaHS*math.pi/2))**2
        sigma_max=math.sqrt(GI_cr*2*kn)
        tau_max=math.sqrt(GII_cr*2*kt)
        mu=2*GII_cr*kt/(tau_cr)**2
        sigma_cr=sigma_max/math.sqrt(mu)
        GI_ten_cr=sigma_cr**2/2/kn

        # Generacion de string con las prop que se copian en el .inp
        Adhesive_prop=str(sigma_cr)+','+str(GI_ten_cr)+','+str(lambdaHS)+','+str(
            kt_kn)+','+str(h_adh)+','+str(mu)
        print('\t\tProp adhesive: '+Adhesive_prop)

        # Copia de las propiedades de esta iteracion en el .inp que tiene
        las prop de la iteracion previa
        PMTESC.cambio_linea_txt(name_INP, Adhesive_prop_previo, Adhesive_prop)
        Adhesive_prop_previo=Adhesive_prop

        # Llamada al PMTESCmain_called.py que es el main original de MAR
        call('python' + ' ' + Main_called + ' ' + str(index1) + ' ' + str(index2),
            shell=True)

end = time.time()
print('La simulacion GLOBAL ha terminado satisfactoriamente en %.2f' %((end-start)
/60)+' min')

```



# Appendix B

## Create an executable to run python scripts

---

To create a *.exe* file from *Python* scripts, although other packages are also possible, the following methodology has been addressed in this project.

In case more than one script is used, they should be able to read the secondary scripts path routes, since the executable will save them in a temporary file (in my case: *C:\Users\andres\AppData\Local\Temp*) and the console must be able to find them. A possible Python function to read those paths is attached in Code B.1:

**Code B.1** Python function to read absolute paths.

```
import os, sys

def resolver_ruta(ruta_relativa):
    if hasattr(sys, '_MEIPASS'):
        return os.path.join(sys._MEIPASS, ruta_relativa)
    return os.path.join(os.path.abspath('.'), ruta_relativa)
```

After this clarification, the following steps should be addressed:

- Install the Python package *pyinstaller* through *pip* in the PC console: *pip install pyinstaller*.
- Gather all the necessary *.py* files (main and those which are called), as well as the *launcher.py* in the same directory.
- Specify the scripts which are called by the main file in the launcher line *'-add-data='*.

In case it would be necessary to import an user-defined library, the lines written in Code B.2 can be used, being *PMTESC* the name of the library and saved as *PMTESC.py*

- In the PC console:
  - Go to the just mentioned directory: *cd C:\ANDRES\PMTE – V15\PMTE – V15 – 4 – exe*
  - Run the launcher: *python launcher.py*. See an example of this launcher in Code B.3.
  - Wait until the PC console shows a success message:  
*50729 INFO: Building EXE from EXE-00.toc completed successfully.*

After these steps, an executable file (*.exe*) will be generated in the *\dist* file. This *.exe* does not required any additional script to call, since we have already included them. In addition, a *.spec* file with the main script name is created. This file summarizes the content of the *.exe*. An example of *.spec* file is shown in Code B.4.

In order to include more arguments, the web page attached in [18] might be consulted.

**Code B.2** Lines to import a library defined by the user.

```
sys.path.insert(1, resolver_ruta('PMTESC.py'))
import PMTESC
```

**Code B.3** Example of *launcher.py*.

```
import PyInstaller.__main__

PyInstaller.__main__.run([
    'PMTESCmain.py', #archivo main
    '--onefile', #agrupa todo en un ejecutable, sin carpetas adicionales
    # '--windowed', #al ejecutar el .exe no visualiza la consola
    '--add-data=PMTESCcriTen_ABQ_carga.py;.',
    '--add-data=PMTESCcriTen_ABQ.py;.',
    '--add-data=PMTESCcritEne_ABQ.py;.',
    '--add-data=PMTESCsalDatos_ABQ_Cdespla.py;.',
    '--add-data=PMTESC.py;.',
])
```

**Code B.4** Example of *main.spec*.

```
# -*- mode: python ; coding: utf-8 -*-

block_cipher = None

a = Analysis(['PMTESCmain.py'],
    pathex=['C:\\ANDRES\\OneDrive - UNIVERSIDAD DE SEVILLA\\PMTE and LEBIM
    at MULTIPLE JOINTS\\Ejecutable\\4_PMTE\\PMTE_exe_script_04'],
    binaries=[],
    datas=[('PMTESCcriTen_ABQ_carga.py', '.'), ('PMTESCcriTen_ABQ.py', '.'),
    ('PMTESCcritEne_ABQ.py', '.'), ('PMTESCsalDatos_ABQ_Cdespla.py', '.'),
    ('PMTESC.py', '.')],
    hiddenimports=[],
    hookspath=[],
    runtime_hooks=[],
    excludes=[],
    win_no_prefer_redirects=False,
    win_private_assemblies=False,
    cipher=block_cipher,
    noarchive=False)
pyz = PYZ(a.pure, a.zipped_data,
    cipher=block_cipher)
exe = EXE(pyz,
    a.scripts,
    a.binaries,
    a.zipfiles,
    a.datas,
    [],
    name='PMTESCmain',
    debug=False,
    bootloader_ignore_signals=False,
    strip=False,
    upx=True,
```



---

```
upx_exclude=[],  
runtime_tmpdir=None,  
console=True )
```



# How to prepare and run PMTE-SC codes

---

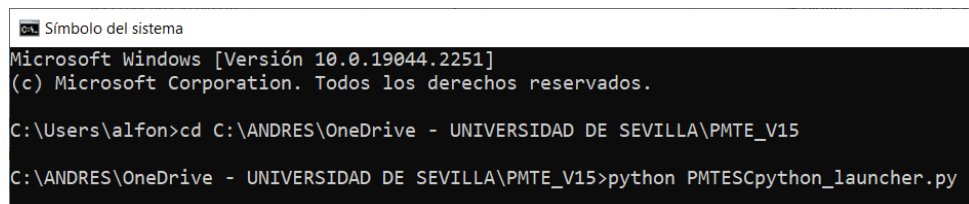
In this appendix, two ways of running PMTE-SC codes will be addressed: from Python 2.7 and from ABAQUS, version 6.14.

## C.1 Launch multiprocessing from Python 2.7

These codes can be launched from Linux (CLUSTER) or Windows (from the PC Command Prompt (*cmd.exe*), which is desirable because it prints messages of secondary scripts and errors; or from an interpreter like *Spyder*, which only prints messages from the main).

### C.1.1 Procedure to launch the codes

- Windows
  - If we are running from an interpreter like *Spyder*, it is only needed to establish the work directory as the folder where the codes are saved and run the main code.
  - If we are running from the Command Prompt (see Figure C.1), it is needed to:
    - \* Open the Command Prompt by writing *cmd* in the Windows bar tool.
    - \* Establish the work directory with the command *cd* and the folder path.
    - \* Run the code calling python and the launcher script: *python PMTESCpython\_launcher.py*.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\alfon>cd C:\ANDRES\OneDrive - UNIVERSIDAD DE SEVILLA\PMTE_V15

C:\ANDRES\OneDrive - UNIVERSIDAD DE SEVILLA\PMTE_V15>python PMTESCpython_launcher.py
```

Figure C.1 Commands to launch from the Command Prompt.

- Linux (Cluster)
  - To run from the CLUSTER, it is needed to:
    - Place all the scripts, input file, *.inp* and UMAT in the *FileZilla* folder *PREPARAPMTESCV15* (see Figure C.2).

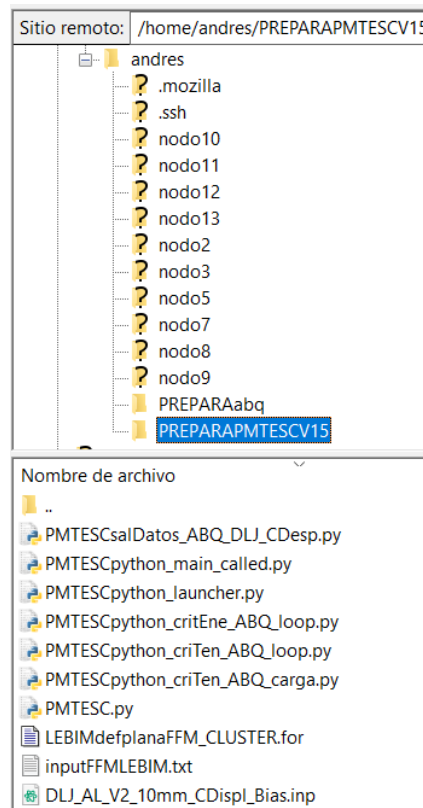


Figure C.2 Files in folder *PREPARAPMTESCV15*.

- Edit the *pythonslurmPMTESCV15.bach* file choosing the #SBATCH options: node where it will be launched, number of CPU's, job-name, which must be the same than the folder of the input file, etc. Also, at the end, it should be written the command to run the main script at the interpreter through python: *python PMTEscpython\_launcher.py >salida.out* (see Code C.1).

Code C.1 Example of *pythonslurmPMTESCV15.bach* file.

```
#SBATCH --nodes=1
#SBATCH --odelist=nodo9
#SBATCH --ntasks-per-node=4
#SBATCH --time=100:00:00
#SBATCH --job-name=DLJ_CCarg
JOB_NAME=$SLURM_JOB_NAME
DIRECTORIO=$JOB_NAME
WORK_DIR=$HOME/$DIRECTORIO
unset SLURM_GTIDS

mkdir -p $WORK_DIR
cd $WORK_DIR
echo "El directorio de trabajo actual es 'pwd'"
echo "-----"
MAQUINA='tail $PBS_NODEFILE -n 1'
echo "Su trabajo se está ejecutando en el $MAQUINA"
echo "-----"
NUM_PROCS=$SLURM_NPROCS
echo "Funcionando en $NUM_PROCS procesador/es."
```

```

echo "La ejecución ha comenzado el: `date`"
echo "-----"

ssh master "cd /home/$USER/PREPARAPMTECV15; tar cfz rts.tgz *"
scp master:/home/$USER/PREPARAPMTECV15/rts.tgz .
tar xzf rts.tgz
rm -f rts.tgz
ssh master rm /home/$USER/PREPARAPMTECV15/rts.tgz

source /opt/intel/oneapi/setvars.sh
export LD_PRELOAD=/opt/intel/oneapi/compiler/2021.1.1/linux/compiler/lib/
intel64_lin/libirc.so

python PMTESCpython_launcher.py >salida.out

echo "-----"
echo "FIN de la ejecución el: `date`"
@

```

– Run from *PUTTY* application with the command *sbach pythonslurmPMTECV15.bach*.

### C.1.2 Procedure to prepare the codes

In the following, the different files will be briefly explained. Furthermore, those which need any modification to be able to perform a desired job will be further detailed to allow the user to easily edit them:

- *PMTESCpython\_launcher.py*: This script is the launcher, which means that we will run this script and it will call the other ones. It is prepared to run several analyses consecutively, which is very useful to run parametric analyses, like the ones done in the inverse problem (see section 5.3). To do that, the properties of the interphase should be specified in the script. Currently, it allows to sweep in  $\tau^{cr}$  and  $G^I$  by changing the *.inp* line where the properties are set in each couple of parameter, but this can be modified to run around other parameters or meshes. The range of the parameters is specified with a *linspace*, so it is needed to specify: minimum value, maximum value and the number of parameters ( $N_{\tau\_cr}$  and  $N_{G^I}$ ). There will be as many combinations as  $N_{\tau\_cr} \cdot N_{G^I}$  and one folder will be created for each of them.

It is important to verify that the initial *.inp* exactly contains the properties of the string *Adhesive\_prop\_previo*, which is '10.0, 1.0, 0.5, 0.25, 0.01, 1.0' (including same spaces) so it is the same string and it is replaced properly.

In case we only want to run one analysis, it is enough to indicate it in the number of parameters that we sweep:  $N_{\tau\_cr}=1$  and  $N_{G^I}=1$ . The properties used will be those corresponding to the minimum value in case the *.inp* file contains the property string to be changed or the ones of the *.inp* file in case it is different.

Briefly said, the script sweeps both parameters changing the properties of the *.inp* file and calling the *PMTESCpython\_main\_called.py* script for each combination.

- *PMTESCpython\_main\_called.py*: this is the main code. The arguments of this script are two indexes that correspond to the parametric analyses so every analysis is run in a different folder.

The only modification needed is to include the arguments when calling the script to write the *archivo\_salida.txt*. This script, which is explained below, should take some variables from the main script and they must be inserted as arguments. An example of the line which calls this script is showed in Code C.2.

**Code C.2** Lines to call the script to write *archivo\_salida.txt*.

```
call('abaqus python'+ ' '+salidaDatos+ ' '+name_files+ ' '+ \
     wdNmin.replace(' ','*')+ ' ' +salida_datos3.replace(' ','*')+ ' '+\
     str(k+1)+ ' '+str(m)+ ' '+str(damageN[0][1])+ ' '+str(damageN[0][0])+ \
     ' '+str(len(damageTen)),shell=True)
```

- *PMTESCpython\_criTen\_ABQ\_carga.py*: this script calculates the *LEBIM load*, which is the load needed so the first element satisfies the stress criteria.

This script does not require any modification.

- *PMTESCpython\_criTen\_ABQ\_loop.py*: this script reads *.odb* files from ABAQUS and determines what elements of the interphase satisfy the stress criteria.

This script does not require any modification.

- *PMTESCpython\_criEne\_ABQ\_called.py*: this script contains everything related to energy calculations. In particular, it determines if the elements that satisfied the stress criteria also satisfy the energetic criteria or not.

This script does not require any modification.

- *PMTESC.py*: this script is a library which contains functions that are useful for the rest of the scripts like copying files, editing subroutines or creating text files.

This script does not require any modification.

- *PMTESCpython\_salDatos\_ABQ\_DLJ.py*: this scripts contains the information to write the output file *archivo\_salida*. Therefore, it is different from one problem to another. The name of the script should be specified in the *inputFFMLEBIM.txt* file, like will be explained below. The variables needed to run the script should be inserted as arguments from the *PMTESCpython\_main\_called.py*, like previously mentioned. This script typically gets results from *.odb* files like displacements, reactions or stresses that are useful to plot graphs and verify the analyses have been performed properly.

- *LEBIMdefplanaFFM.for*: this script is the UMAT (user-material) subroutine which includes the *LEBIM* theory. It is written in FORTRAN language. This language is very strict with the spacing, so much attention should be paid around this issue. The extension of the file should be *.for*.

This file only requires to modify the path of the file *datos\_procesar.txt*, which will contain the damaged elements. Since this file appears in the folder where the calculations are done, the required path is the working directory. Code C.3 shows an example of the lines that define the path for Windows and Cluster. Comments are made after letter 'c'. When running from the CLUSTER, the user name should be indicated; whereas when running from Windows, it should be the working directory path of the PC. For both cases, the folder called *NAMEFOLDER* should be maintained so it is automatically changed from the folder name inserted in the *inputFFMLEBIM.txt* file. With this in mind, when running from Cluster, the UMAT file does not need any modification.

**Code C.3** Path to call *datos\_procesar.txt* in subroutines.

```
c path for Windows
      fullname='C:\ANDRES\OneDrive - UNIVERSIDAD DE SEVILLA\'
      *// 'PMTE and LEBIM at MULTIPLE JOINTS\DLJ_AL\A09_Graf_Energia\'
      *// 'NAMEFOLDER\'
      *// 'datos_procesar.txt'
```

```

c path for Cluster
    fullname='/home/andres/'
    *//'NAMEFOLDER/'
    *//'datos_procesar.txt'

```

- *inputFFMLEBIM.txt*: this text file contains the input parameters needed to run the code depending on the preferences of the user. It consists on several pair of lines where the first one is the description of the parameter and the second one is the value of the parameter itself:

- Name of the *.inp* file, without extension.
- Name of the working directory. It will be used to complete the path of the file *datos\_procesar.txt* in the UMAT file.
- Name of the UMAT file, without extension.
- Maximum number of iterations (*num\_iteraciones*).
- Name of the interface set in the *.inp* file.
- String with the parameter name and its initial value, which will change its value each iteration.
- First value of the load.
- Parameter name.
- Value of the initial increment.
- String to indicate the model that is analysed. This string is read by the function *set\_BC(name\_INP, TestModel, control, carga)* in the *PMTESCabq\_User.py* library where the boundary conditions are detailed for each model and type of control. This string is only used when launched from ABAQUS.
- Python file name (including extension *.py*) in charge of the creation of the output file.
- Variable to indicate whether to save *.odb* files or not.

$$\left\{ \begin{array}{l} 0 \text{ If it saves nothing} \\ 1 \text{ If it saves every } .odb \text{ file} \\ 2 \text{ If it saves only } n0j0.odb \text{ file for each k,m step} \end{array} \right. \quad (\text{C.1})$$

- Integer to indicate type of control.

$$\left\{ \begin{array}{l} 0 \text{ Displacement control} \\ 1 \text{ Load control} \end{array} \right. \quad (\text{C.2})$$

- Factor to multiply the LEBIM load so not a single element satisfies the stress criteria in the first step. Typical value: 0.9.
- Maximum number of *m* steps.
- Binary integer to indicate whether to always increase load or to be able to caught snap-backs.

$$\left\{ \begin{array}{l} 0 \text{ Allow snap-backs} \\ 1 \text{ Always increase load} \end{array} \right. \quad (\text{C.3})$$

- (adapative): binary integer to indicate whether to use the adapative model from A. Laguna or the original from M. Muñoz-Reja.

$$\left\{ \begin{array}{l} 0 \text{ Original from M. Muñoz-Reja} \\ 1 \text{ Adapative model from A. Laguna} \end{array} \right. \quad (\text{C.4})$$

- (parada\_rotura): binary integer to indicate wheter to exit the code when first failure is reached or continue.

$$\begin{cases} 0 & \text{Continue} \\ 1 & \text{Exit} \end{cases} \quad (\text{C.5})$$

- (fractura\_infinitesimal): binary integer to indicate if it is desired an infinitesimal failure, which means to make small increments so every failure only involves 1 element, or not.

$$\begin{cases} 0 & \text{No restriction in number of damaged elements} \\ 1 & \text{Infinitesimal growth, only failures of 1 element allowed} \end{cases} \quad (\text{C.6})$$

- (TOL): float that indicates the desired tolerance for the adaptative model. It should be contained in (0,1] interval. In case, it is not wanted to reduce increments with the tolerance criteria, it can be set to higher values than 1 and it will not be taken into account.
- (parada\_ante\_carga): float to indicate wheter to exit the code when a certain load is reached.

$$\begin{cases} 0 & \text{No matter the load, it continues} \\ \text{Float} & \text{Load at which the code will exit} \end{cases} \quad (\text{C.7})$$

- (incre\_factor): float factor that multiplies the increment to load faster. For value equal to 1, it maintains constant.

## C.2 Launch from ABAQUS (Vers. 6.14)

In this case, the scripts can only be launched from the CLUSTER because when it is done from .cae in the PC it might happen that the *datos\_procesar.txt* file is renamed with the PC name (for example *datos\_procesar-DESKTOP-M5IDNAH.txt*) so the UMAT is not able to read it. This situation seems not to appear when the scripts are run in the CLUSTER.

### C.2.1 Procedure to launch the codes

The procedure to launch the codes from the CLUSTER are similar to the previous one:

- Place all the scripts, input file, *.inp* and UMAT in the *FileZilla* folder *PREPARAPMTESCV15* (see Figure C.2).
- Edit the *abaqusslurmPMTESCV15.bach* file choosing the #SBACH options: node where it will be launched, number of CPU's, job-name, which must be the same than the folder of the input file, etc. Also, at the end, it should be written the command to call ABAQUS and launch the script: */opt/abaqus/Commands/abq6141 cae noGui=PMTESCabq\_launcher.py > salida.out*. Notice that this line is the only one that makes the *.bach* different compared to the file of the multiprocessing because it is run from ABAQUS instead of Python.
- Run from *PUTTY* application with the command: *sbach abaqusslurmPMTESCV15.bach*.

### C.2.2 Procedure to prepare the codes

The files required are essentially equivalent to those explained above, with some differences:

- There is an additional library called *PMTESCabq\_User.py* where the boundary conditions and output file are detailed depending on the model it is analysed. This library is included because those aspects vary from one problem to another, so it is designed to be easily updated by the user. Some explanations around these functions will be made:



– Function *set\_BC*

This function is created to establish the boundary conditions. The current arguments are: the name of the INP, the model that is analysed, the type of control and the current load. For each test model and type of control, different boundary conditions are established by means of conditional control structures. For new models, new conditions should be established. Currently, there are two possibilities: 'DCB' and 'DLJ'. These string names are inserted in the INPUT file.

In order to know the commands to implement the boundary conditions, the use of macros is recommended. This is, starting with the model opened in .CAE, select File - Macro Manager - Create - Work directory - Continue. After this, everything that is done in the model will be saved in a file until selecting the stop button, so if the boundary conditions are established, we will be able to know the Python commands. Once we know them, we should adjust the parameters (generally stringers and floats) to those of our codes: INP name, instances names, set names, load values, etc.

In case other arguments are required, they should be included in the main script.

– Function *sal\_Datos*

In a similar way, this function is made to create the output file, which depends on the model is going to be analysed. It is called from the main script where the first argument is the string of the python script name used when it is run from Python. This string is used to enter in the condition corresponding to the model we are currently running. Each condition calls a different function, with the same name of the string (without the extension *.py*), that is defined below, where the commands to create the output file are written. These commands can be known from the *.odb* file by using the Macro Manager like explained above.

In case other arguments are required, they should be included in the main script.

- The UMAT is the same than the one used to run from Python with the only difference of the extension of the file *LEBIMdefplanaFFM.f*, which is, in this case, *.f*.

Finally, a consideration might be presented: the results might slightly differ when running from Python or ABAQUS because the number of significant digits change. Anyways, this is not worrying because the differences are very small.



# List of Figures

---

2.1	Interface modelled through spring distribution [22]	4
2.2	Portion of the linear-elastic layer under normal (a) and tangential (b) stresses [20]	5
2.3	Normal (a) and tangential (b) linear elastic-brittle law for undamaged interface [23]	5
2.4	Normal (a) and tangential (b) linear elastic-brittle law for broken interface [23]	6
2.5	CCFFM+LEBIM constitutive law in pure fracture (left) mode I and (right) mode II [12]	9
2.6	Examples of application of PMTE-SC. Total energy vs crack length [11]	10
3.1	Benchmark problem: Double Cantilever Beam test [15]	11
3.2	Interface casuistry. (a) Undamaged points of interface $\Gamma$ ; (b) points satisfying the stress criterion ( $A_\sigma \subset \Gamma$ ) [15]	12
3.3	Possible configurations of initial subsets [15]. (a) empty set; (b) full set; (c) intermediate case with damaged and undamaged points	13
3.4	Nomenclature used in pseudocode [15]	14
3.5	General pseudo-code for an Alternating Minimization Algorithm for PMTE-SC [15]	16
3.6	Analytical and numerical results for DCB example from [15] under displacement control. Total energy vs crack length. (Steps 1 and 2)	17
3.7	Analytical and numerical results for DCB example from [15] under displacement control. Total energy vs crack length. (Steps 1, 2 and 3)	17
3.8	Analytical and numerical results for DCB example from [15] under load control. Total energy vs crack length. (Steps 1 to 5)	18
3.9	Analytical and numerical results for DCB example from [15] under load control. Total energy vs crack length. (Steps 1 to 13)	19
4.1	DCB sample	21
4.2	Nomenclature for the beam nodes. In red for upper beam and yellow for lower beam	22
4.3	Boundary condition at reference nodes of the beams	23
4.4	Nodes' displacements for the analysis where the load is applied in the centroid	23
4.5	Parameter study: step increment	24
4.6	Mesh convergence study. Displacement control	25
4.7	Mesh convergence study. Load control	26
4.8	High accuracy model - Load control ( $\Delta F = 5N$ )	28
4.9	High accuracy model - Displacement control ( $\Delta U = 0.01mm$ )	28
4.10	Adaptative model flowchart	30
4.11	Original model - Displacement control	31
4.12	Adaptative model - Displacement control	32
4.13	Original vs Adaptative model - Displacement control	33

4.14	Total energy as function of crack length. Displacement control	35
4.15	Total energy as function of crack length. Load control	35
5.1	DLJ sample	37
5.2	Timoshenko and Euler-Bernoulli beams	38
5.3	Numerical models	39
5.4	Upper and lower stresses path. Normal and shear stresses	40
5.5	Isotropic overlapped model. Numerical (El. size: 0.125mm and 0.03125mm) vs Analytical ( $K_z = TY = 1.2$ and $K_z = TY = 0$ )	40
5.6	Mesh of the specimen for DLJ test	41
5.7	First analytical DLJ models. Figures from [2]	42
5.8	DLJ model of Volkersen and de Bruyne in 1944. Parameter influence. (From lesson documentation from A. Barroso)	42
5.9	Orthotropic models. Numerical (0.125mm) vs Analytical ( $K_z = TY = 1.2$ and $K_z = TY = 0$ )	43
5.10	Composite adherends from [19]. Shear and peeling stresses	44
5.11	Experiments from [2]: Tests 1D	45
5.12	Compilation of experiments from A. Barroso ([2]): Test 1D	45
5.13	Properties for adhesive Araldite AV138 [3]	46
5.14	Experiments for overlap length of 5mm [3]	46
5.15	Determination of parameter $k_t$ from numerical tool and experiments [3]	47
5.16	Experiments for overlap length of 5mm, 10mm and 20mm [3]	49
5.17	Estimation of parameters $G_{II}^{cr}$ and $\tau_{cr}$ [3]	50
5.18	Experiments for overlap length of 20mm [3]	51
5.19	<i>Curve Fitter</i> MATLAB tool	52
5.20	Surface interpolation with <i>Curve Fitter</i> MATLAB tool	52
5.21	L=5mm - Failure surface: numerical vs experimental (min., mean and max.)	56
5.22	L=10mm - Failure surface: numerical vs experimental (min., mean and max.)	57
5.23	L=20mm - Failure surface: numerical vs experimental (min., mean and max.)	57
5.24	L=5mm - Failure surface and intersection curves	58
5.25	L=10mm - Failure surface and intersection curves	59
5.26	L=20mm - Failure surface and intersection curves	59
5.27	Intersection curves at mean values of experiments for 3 overlap lengths	60
5.28	Intersection curves at min., mean and max. values of experiments for 3 overlap lengths	60
5.29	Valid regions of $\tau_{cr}$ and $G_{II}^{cr}$ for 3 overlap lengths	61
5.30	PMTE-SC numerical results vs experiments. Reaction with respect to displacement for an overlap length of 5mm	62
5.31	PMTE-SC numerical results vs experiments. Failure load for every overlap length	62
5.32	Numerical curves of energetic behaviour for overlap length of 10mm under load control	64
5.33	Numerical breakage of adhesive under load control	65
5.34	Stresses at adhesive	65
5.35	Numerical curves of energetic behaviour for overlap length of 10mm under displacement control	66
6.1	Evolution of PMTE-SC codes	67
C.1	Commands to launch from the Command Prompt	79
C.2	Files in folder <i>PREPARAPMTESC15</i>	80

# List of Tables

---

4.1	Reaction forces from analyses where displacement is applied in different points	23
4.2	Mesh convergence study. Size details	25
4.3	Comparison between different <i>PMTE-SC</i> code versions	27
4.4	Max. load and number of steps required to load until first failure for original and adaptative models	32
4.5	Computational cost and number of steps required to reach 0.15mm for original and adaptative models	32
5.1	Interface properties for different tests used in [13]	47
5.2	Interface properties	48
5.3	Critical parameters of first approach	49
5.4	Failure loads [N] for tests from [3]	51



# Bibliography

---

- [1] G. A. Arnold, *VI. the phenomena of rupture and flow in solids*, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character **221** (1921), no. 582-593, 163–198.
- [2] A. Barroso, *Caracterización de estados singulares de tensión en esquinas multimateriales. Aplicación a uniones adhesivas con materiales compuestos*, Ph.D. thesis, Departamento de Mecánica de Medios Continuos y Teoría de Estructuras. Universidad de Sevilla, May 2007.
- [3] R. Campilho, M. Banea, A. Pinto, L.F.M. Silva, and A. De Jesus, *Strength prediction of single and double-lap joints by standard and extended finite element modelling*, International Journal of Adhesion and Adhesives **31** (2011), 363–372.
- [4] Zvi Hashin, *Finite thermoelastic fracture criterion with application to laminate cracking analysis*, Journal of the Mechanics and Physics of Solids **44** (1996), no. 7, 1129–1145.
- [5] J.W. Hutchinson and Z. Suo, *Mixed mode cracking in layered materials*, Advances in Applied Mechanics, vol. 29, Elsevier, 1991, pp. 63–191.
- [6] G. R. Irwin, *Analysis of stresses and strains near the end of a crack traversing a plate*, Journal of Applied Mechanics **24** (1957), no. 3, 361–364.
- [7] Dominique Leguillon, *Strength or toughness? a criterion for crack onset at a notch*, European Journal of Mechanics - A/Solids **21** (2002), no. 1, 61–72.
- [8] Hart-Smith LJ, *Adhesive-bonded single-lap joints*, Douglas Aircraft Co., NASA Langley Report CR 112236, 1973.
- [9] A. López, *Estudio numérico del daño en interfases de materiales compuestos a través de LEBIM y CCFM*, Ph.D. thesis, Departamento de Mecánica de Medios Continuos y Teoría de Estructuras. Universidad de Sevilla, 2021.
- [10] V. Mantič, *Interface crack onset at a circular cylindrical inclusion under a remote transverse tension. application of a coupled stress and energy criterion*, International Journal of Solids and Structures **46** (2009), no. 6, 1287–1304.
- [11] Vladislav Mantič, *Prediction of initiation and growth of cracks in composites. coupled stress and energy criterion of the finite fracture mechanics*, (2014).
- [12] M. Muñoz-Reja, *Analytical and numerical predictions by LEBIM and FFM for interface cracks in composites and their joints*, Ph.D. thesis, Departamento de Mecánica de Medios Continuos y Teoría de Estructuras. Universidad de Sevilla, May 2021.

- [13] M. Muñoz-Reja, P. Cornetti, L. Távara, and V. Mantič, *Interface crack model using finite fracture mechanics applied to the double pull-push shear test*, International Journal of Solids and Structures **188-189** (2020), 56–73.
- [14] M. Muñoz-Reja, V. Mantič, and L. Távara, *Comparative analytical study of the coupled criterion and the principle of minimum total energy with stress condition applied to linear elastic interfaces*, Theoretical and Applied Fracture Mechanics **119** (2022), 103274.
- [15] M. Muñoz-Reja, V. Mantič, L. Távara, and C. Panagiotopoulo, *Numerical implementation of the ccfrm applied to lelim by pmte*, "UNDER REVIEW".
- [16] M. Muñoz-Reja, L. Távara, V. Mantič, and P. Cornetti, *A numerical implementation of the coupled criterion of finite fracture mechanics for elastic interfaces*, Theoretical and Applied Fracture Mechanics **108** (2020), 102607.
- [17] Ludwig Prandtl and W. G. Knauss, *A thought model for the fracture of brittle solids*, International Journal of Fracture **171** (2011), no. 2, 105–109.
- [18] PyInstaller contributors, <https://pyinstaller.org/en/stable/usage.html#options>.
- [19] M.Y. Tsai and J. Morton, *An investigation into the stresses in double-lap adhesive joints with laminated composite adherends*, International Journal of Solids and Structures **47** (2010), no. 24, 3317–3325.
- [20] L. Távara, V. Mantič, E. Graciani, and F. París, *Bem analysis of crack onset and propagation along fiber–matrix interface under transverse tension using a linear elastic–brittle interface model*, Engineering Analysis with Boundary Elements **35** (2011), no. 2, 207–222.
- [21] L. Távara, L. Moreno, E. Paloma, and V. Mantič, *Accurate modelling of instabilities caused by multi-site interface-crack onset and propagation in composites using the sequentially linear analysis and abaqus*, Composite Structures **225** (2019), 110993.
- [22] L. Távara, J. Reinoso, D. Castillo, and V. Mantič, *Mixed-mode failure of interfaces studied by the 2d linear elastic-brittle interface model: macro- and micro-mechanical finite element applications in composites*, The Journal of Adhesion (2017).
- [23] V. Vladislav, L. Távara, A. Blázquez, E. Graciani, and F. París, *A linear elastic-brittle interface model: application for the onset and propagation of a fibre-matrix interface crack under biaxial transverse loads*, International Journal of Fracture **195 (1)** (2015), 15–38.