AALBORG UNIVERSITY

STUDENT REPORT

DESIGN OF MECHANICAL SYSTEMS

MASTER THESIS

Design of an Experimental Procedure for the Characterization of Cohesive Laws in Composites

Participants
Félix Prieto Viejo
José Mato Sanz
Pere Joan Jaume Camps

Supervisor
Brian Lau Verndal Bak

3rd of June 2020

AALBORG UNIVERSITY
STUDENT REPORT

4. semester - School of Engineering and Science
Design of Mechanical Systems
Fibigerstræde 16
9220 Aalborg East
http://www.ses.aau.dk

Title:
Design of an Experimental Procedure for the Characterization of Cohesive Laws in Composites
Theme:
Cohesive Zone Modelling, Fracture Mechanics, Monte Carlo Methods, Inverse Problems, Experimental Mechanics
Semester:

DMS4, Spring semester 2020

Semester theme:

Master Thesis

Project period:

February 3rd - June 3rd 2020

ECTS:

30

Project group:

DMS4 - 23G

Participants:

Félix Prieto Viejo
José Mato Sanz
Pere Joan Jaume Camps

Supervisor:

Brian Lau Verndal Bak

External examiner:

Michal Kazimierz Budzik

Page numbers: 139
Appendix enclosures: 9
Date of completion: June 3rd, 2020

Abstract:

Through this report the design and validation of an experimental procedure that aims to characterize cohesive laws of layered composite materials in an accurate and simple way is done. The cohesive law determination is done from a displacement field of one arm of a DCB specimen via an inverse parameter identification procedure developed in the previous semester. Studies on the effect of uncertainties related to the experimental procedure on the variability of the cohesive law calculations are performed using the Monte Carlo approach. Correlation studies are performed to identify the parameters that drive the size of the confidence intervals obtained from the Monte Carlo analyses. From the results obtained, modifications in the formulation of the inverse parameter model are introduced to improve the accuracy and stability of the cohesive law obtained.
Based on previous experience and the results from the statistical analysis performed, an experimental procedure is designed to reduce the impact of the variables that have been found to strongly affect the solutions obtained. The experimental procedure is based on DIC and only requires displacement fields to be measured.
The motivation for this project is to develop an experimental method that can be widely applicable in industry thanks to its speed and simplicity, that eliminates the human interpretation on the data treatment process and which has the possibility of being applicable to fatigue loading.

Preface

This master thesis presents the findings, discussion and conclusions of the group Fib.14-23G, during the 4th semester of the master's program in Design of Mechanical Systems, at Aalborg University. The project, as a continuation of the 3rd semester work, started the 3rd of February 2020, and concluded the 3rd of June 2020. During this period, the COVID-19 outbreak, which has affected the entire AAU organization, has also affected the experimental part of this thesis.

Together with this document, a compressed folder containing the MATLAB scripts and experimental data, used to obtain the results of the present work is attached.

The authors acknowledge the supervisor of this thesis, Associate Professor Ph.D. Brian Lau Verndal Bak, for its comprehensive evaluation and patience during the whole year. His guidance and suggestions, have helped the development of the group as students and engineers. A special gratitude is also given to the staff from Materials and Production workshop at Aalborg University. Last but not least, thanks to our families, for all the support given.

Reading guide:

Due to the amount of information gathered in this master thesis, the theory on which this project is based is displayed in Appendices A, B and C, and the main body of the report is left for presenting the work done. Therefore, it is strongly recommended to read the aforementioned appendices, for a full understanding of the information exposed.

It is also recommended to read Appendix D, to get an overview of the work of the previous semester project Viejo et al. [2019].

Signatures

Félix Prieto Viejo

José Mato Sanz

Pere Joan Jaume Camps

Table of contents

1 Introduction

In this chapter an overall idea of the background, motivation and state-of-the art for the developed methodology is presented. Then, as this project is a continuation of a previous semester (third semester) project, a clear definition between what has been done and what is aimed to be done is established. Finally, gathering all the previous information given, the problem statement of this project is formulated. Additionally, an example of a solution obtained using the developed methodology is presented.

1.1 Contextual Background

The use of composite materials in industry due to their advantages against other materials has been increasing in the last years. The failure characterization of these type of materials or the prediction of component life have been an issue since their introduction, especially due to their applications in aerospace, aeronautic, automotive or wind turbine industry, where a failure can lead to catastrophic consequences [King, 1989]. If the physical limits of a material are well characterized under real operational conditions, their expected life could be better defined, maintenance tasks would be planned only when necessary and failure of structures and mechanisms could be avoided or foreseen. This would reduce the negative economic impact that most of the companies experience when an unexpected failure on one of their products happen.

Another issue is that most of the failure techniques that try to characterize the behaviour of these materials to failure are not simple to apply. They often require complex equipment designed especially with a purpose and are subjected to the judgement and interpretation of the human being evaluating the case. This, is just for the quasi-static case. Fatigue is an even harder area of research that requires more effort and knowledge, and where some of the methods that are currently used in industry to assess delamination behaviour fail to perform [Sørensen, 2010].

Whether all these factors mentioned can be a reality or not, rests on the importance of research and the establishment of methodologies that are based on real life experiments and that are easy to implement in reality. The objective of this project is to elaborate a methodology that effectively characterizes the quasi static delamination behaviour of layered composites in a fast and simple way, without being subjected to the user's interpretation, that is validated for real-life experimental conditions and could potentially be used for fatigue analysis of components.

1.2 Motivation

The use of composite materials plays an essential role in industry nowadays, and its use has increased since they were introduced in the mid-twentieth century. The word *composite*, refers to a combination of two or more materials to create a useful third material [Jones, 1999]. The ability of composite materials to be tailored to fit specific applications supports their popularity in many different industrial areas. This project centers its investigation in *layered composites*, which are formed by fibrous layers stacked up together.

With the introduction of composite materials, unprecedented failure mechanisms appeared, that recall for new failure theories that account for these novel effects. Among the different failure mechanisms present, such as kink-band failure, fiber-matrix debonding, matrix yielding/failure, fiber micro-buckling or fiber failure, delamination (see Figure 1.1) is the most common and one of the most critical failure mechanisms [Lindgaard and Bak, 2019]. Delamination is defined as the *"development of a crack along the interface between plies"* by Turon et al. [2006]. This failure is mainly caused by the presence of high interlaminar stresses together with a very low through-thickness strength that is characteristic of layered composite materials. Fibers are aligned in a plane to reinforce the structure in the corresponding direction, and when loads are applied perpendicular to the plane of the aligned fibers, the matrix material must carry that load alone. Matrix materials are usually considerably weaker than fibers, and therefore, failure is prone to happen under these conditions.

Figure 1.1: Delamination along the interface in a fibrous laminate composite material [Viejo et al., 2019].

Delamination can be caused by numerous reasons e.g. as a consequence of other failure mechanisms (fiber/matrix debonding), due to defects in the manufacturing processes, or can be induced by geometric features such as notches or free edges (which produce high interlaminar stresses). The possibility of failure due to delamination occurring in composite materials is a known design limit, and depicts a clear obstacle in the usage of this type of material.

To exploit the full potential of composite materials, this knowledge gap of failure mechanisms effect on the performance of the materials must be correctly addressed and investigated. This fact motivates the development of the current project, which objective is to provide a tool that can be used to characterize in a realistic way the aforementioned failure mechanism described in the previous paragraph.

1.3 Theoretical Background and State-of-the-art

Clasically, failure assessment and crack propagation have been analysed via fracture mechanics. Fracture mechanics deals with the irreversibility of the rupture process due to crack nucleation/growth when a load is applied. Griffith [1920] proposed in 1920 an energy criterion based on the energy release rate to explain crack growth. After him, Irwin [1957] reformulated the energy criterion introducing the stress intensity factors using Linear Elastic Fracture Mechanics (LFEM). This knowledge was further extended to develop a series of techniques that are the most used options for the analysis of crack propagation e.g. the J-integral, or the Virtual Crack Closure Technique (VCCT). Nonetheless, the use of LEFM is limited due to the simplicity of its main assumptions:

1. The assumption of an infinitely sharp crack tip leads to infinite stresses at the crack tip.
2. LEFM does not account for crack formation, which implies that a crack must already exist in the body and its location and size are known. It is only valid as a propagation criterion.
3. LEFM is accurate when the nonlinearities are enclosed in a small region around the crack tip. Thereby, materials that exhibit an R-curve behaviour are not correctly represented.
4. LEFM is not suited for numerical method implementation e.g. Finite Element Method (FEM), as its formulation leads to difficulties in the calculations of some fracture parameters e.g. energy release rates, when progressive crack propagation is involved [Turon et al., 2006].

The reasons stated above (1 to 4) suggest the necessity of a theory that can overcome these issues and can represent delamination failure accurately. The solution is found in the Cohesive Zone Modelling (CZM), a theory embedded in the Damage Mechanics framework that assumes that crack faces are brought together by tractions acting against the opening of the crack, which eliminates the problem of the stress singularity at the crack tip [Barenblatt, 1962]. These tractions at the interface represented in Figure 1.2 are directly related to the crack opening displacement by a constitutive relation denominated *Cohesive Law*. The Cohesive Law relates point-wise the separation of the faces and the aforementioned traction which brings the faces together to depict the behaviour of the structure while damage is occurring. In the deformed state, the cohesive tractions, located within the Cohesive Zone (see Figure 1.2) prevent the crack from propagating i.e. the external forces have to overcome the cohesive tractions for the delamination to progress. When the damage is fully developed, there is a complete separation of the crack faces and therefore the crack tip advances further. This cohesive law can be seen as a function that is a material characteristic and can have different shapes (trapezoidal, linear, exponential, etc.) describing materials with different behaviours. Moreover, the CZM uses the strength criterion as a crack initiation criterion [Hillerborg et al., 1976], consequently, it can account for crack propagation and initiation simultaneously. Another advantage of the CZM is its simplicity to be introduced in the FEM, which makes use of interface elements to simulate delamination behaviour.

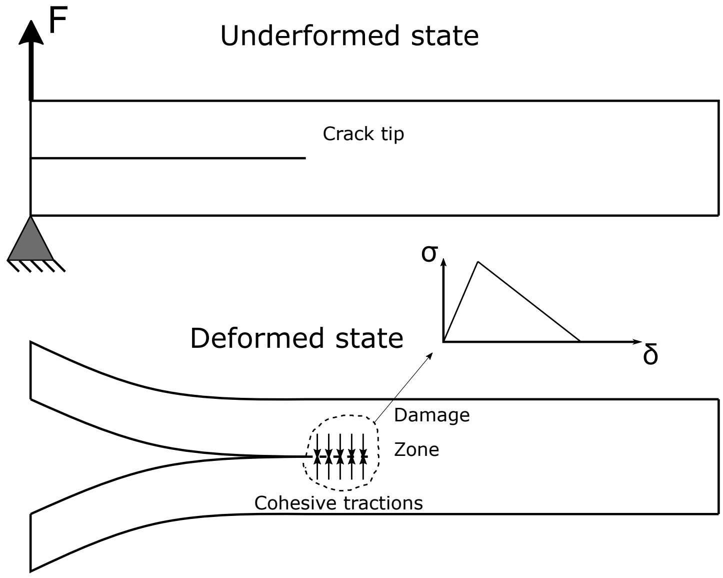


Figure 1.2: Cohesive stresses and bilinear traction-separation relation in a Double Cantilever Beam (DCB) under delamination.

There are several approaches for characterizing cohesive laws. The most simple method to use would be the direct tension test, in which the traction-separation law would be calculated directly from the experiment measuring the stress (σ) and the separation of the crack faces (δ) [Shen et al., 2010]. However, this experiment is very difficult to execute in a perfect way, as it is usual that the results are discarded due to the specimen developing multiple cracks, the rotation in the crack faces of the specimen or the multiple cracks overlapping [Elices et al., 2002] (see Figure 1.3).

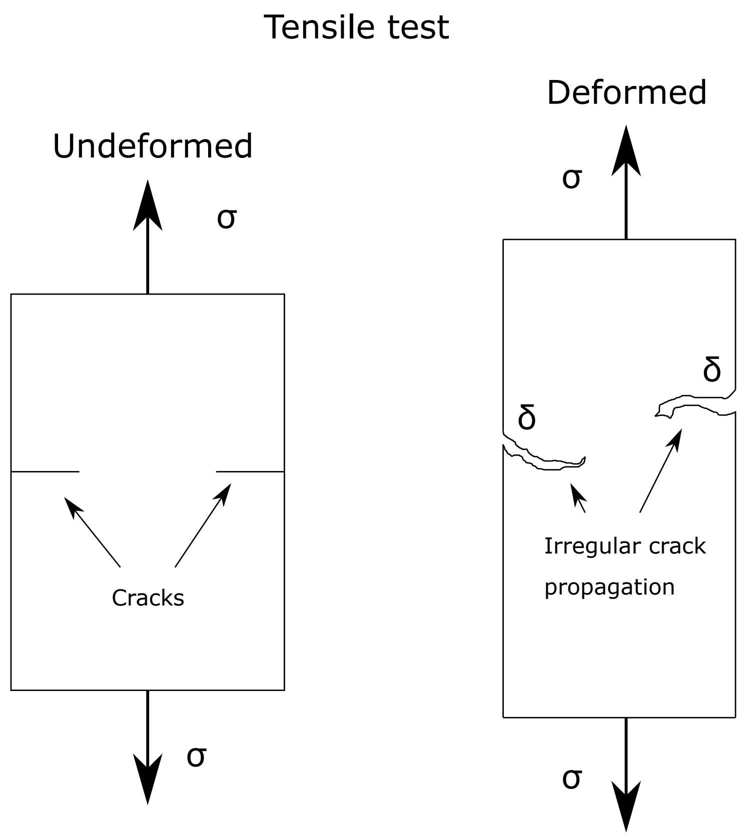


Figure 1.3: Direct tensile test and uneven crack propagation in a tensile specimen.

Other direct method that has been proved to be effective in cohesive law characterization is the method based on the J-integral. This procedure first introduced by Li and Ward [1989] for characterizing the tension-softening relations in cementitious composites is later also used by Sørensen [2010] for determining bridging laws in laminated composite materials. This method relies on the path independent J-integral [Rice, 1968] and its closed-form solutions for particular specimens under particular loading conditions. This method involves measurements of global structural responses to compute the value of the J-integral and the crack tip opening displacement, together with a numerical differentiation of the J-integral to obtain the cohesive stresses (see Figure 1.4).

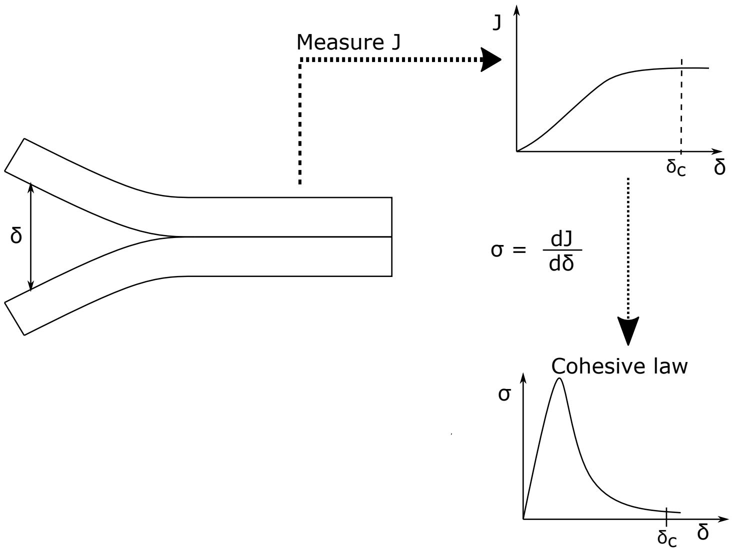


Figure 1.4: J-integral method workflow for the characterization of a cohesive law.

Indirect methods use inverse analysis and the assumption of a shape for the traction-separation relation with a few model parameters [Shen et al., 2010]. For this approach the goal is to determine these parameters, via experimental procedure or numerical simulation. In Ortiz and Pandolfi [1999], an irreversible exponential law is proposed to be implemented in a 3D FE model of a dynamic fracture test, measuring the crack opening and the crack growth velocity. Turon et al. [2006] develops a delamination-initiation criterion for mixed-mode delamination, that is later proved by FE simulations and experimental results. In this case, there is an *a priori* assumption of a bilinear cohesive law (for simplicity), and the parameters to be determined are the load applied and the displacement of the crack tip for different mode mixity ratios.

The assumption of a cohesive law shape is demonstrated to affect the results of the CZM calculations, as stated by Shah et al. [1995] *"the local fracture behaviour is sensitive to the selection of the shape of the cohesive law"*. For this reason, there are other type of indirect methods that mitigate this problem by using inverse identification. Shen et al. [2010] uses an approach that relies on the Digital Image Correlation (DIC) error range and the FEM to simulate a real displacement field and inverse optimization to get a feasible cohesive law. Another approach that eliminates the *a priori* assumption is the one proposed by Jensen et al. [2019], where a multilinear cohesive law is solved using inverse parameter identification. This approach uses measures of global structural responses of moment applied and rotation of the arms of a Double Cantilever Beam (DCB) and a FE simulation for the optimization. The method is then validated by means of the J-integral technique.

This review shows how the vast majority of previous investigations are supported either by FE simulations or by experiments in which global structural responses are measured. FE simulations that involve the implementation of an interface element and experimental procedures involving different loading conditions and a wide variety of specimens depending on the pursued outcome.

The present masters thesis aims to accurately characterize a cohesive law using an indirect method that uses local response measurements and aims to compute interface tractions instead of normal stresses. This is a novel method when compared to existing techniques and requires a thorough investigation of its behaviour and capabilities under real experimental conditions. A more detailed discussion of the work contained in this master thesis is presented in the next section.

1.4 Problem Formulation

The scope of this masters thesis is to study and design an effective experimental methodology for characterization of cohesive laws of composite materials under quasi-static mode I delamination for real experimental conditions. The methodology relies on the use of DIC as measurement method, cohesive zone modelling and beam theory aided by FE modelling. This aims to achieve a methodology that has a wide and simple application and that effectively accounts for crack initiation and propagation under real life working conditions. The main challenge of this masters thesis is to design an experiment that gives a robust and stable response to the beam-based model that has been developed previously, by modifying the model to comply with experimental effects that need to be accounted for to have an accurate description of reality.

This master thesis is a continuation of the work done in the previous semester project Viejo et al. [2019]. An outline is described here to clarify what has already been done (third semester of the masters in Design of Mechanical Systems) and what is done in this final semester of the masters (fourth).

1.4.1 Third Semester (Previous)

A resume of the work that has been carried out in the previous semester project [Viejo et al., 2019] is given here. Further information is found in Appendix D or in Viejo et al. [2019]. The aim of the project is to study and develop a method (based on a beam model approach) for the characterization of the cohesive law of a laminated composite material under quasi-static mode I delamination. The method relies on the approximation of a DCB specimen under delamination to a clamped beam with an unknown distributed load applied. This unknown traction field represents the interfacial tractions in the cohesive zone, which are the unknown to be calculated. The methodology (see Figure 1.5) was formulated based on 2D Finite Element (FE) simulations of a DCB specimen under mode I delamination with balanced moments applied at the end of the arms of the specimen.

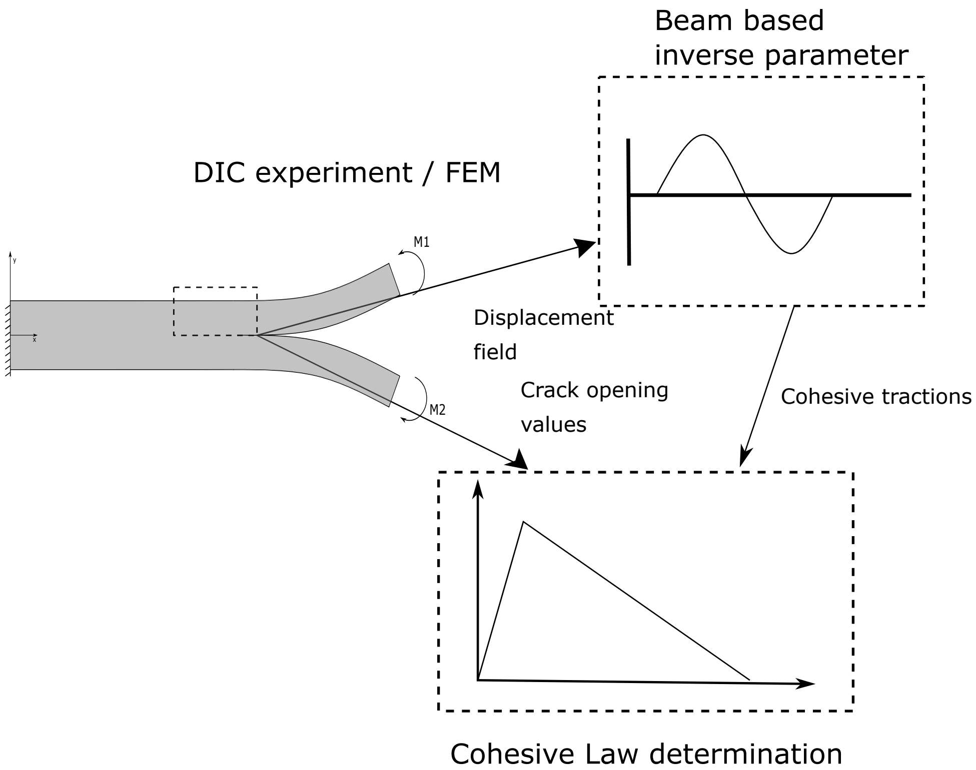


Figure 1.5: Sketch of the methodology developed.

The main purpose for developing a methodology using the presented approach is to avoid some of the difficulties that the present theories about cohesive law characterization. Precisely, due to the fact that the considered best method in the present (the J-integral method) cannot be used for cyclic load applications.

The methodology developed is considered as novel because it is based on a change of approach that is believed to have potential to overtake the methodologies existing in the present. Instead of measuring parameters that are directly related to the cohesive law, the interface tractions are chosen to be calculated. With this change of approach, the flexibility when predicting the shape of the cohesive law calculated is higher when compared to some of the actual cohesive law characterization methods. An inverse parameter identification is used to find a solution for the traction field. This approach (inverse parameter identification) has proven to give reliable results in previous research e.g. Jensen et al. [2019]. The fact that tractions are being predicted as a linear combination of loads allows the problem to be solved using ordinary least-squares, with a guaranteed global minimum, eliminating the nonlinearities that the cohesive law parameter prediction has. Finally, it is decided that the DIC method is used to measure the displacement field of the DCB, so that the local behaviour of the cohesive zone can be captured by the experimental measurements.

The algorithm created (explained in detail in Appendix D) is based on a least-squares optimization technique that aimed to reduce the difference between the displacement caused by the predicted distributed loads in the model and the input displacement field from FE simulations or DIC measurements. The prediction of the interface stresses in the simple beam is formulated first with Euler-Bernoulli beam theory and validated with analytical solutions and later, with a 2D FE model of the experimental procedure. Due to the limitations that Euler-Bernoulli has in the behaviour of a beam, it is decided to improve the formulation of the model using Timoshenko beam theory. In addition, a complete validation involving analytical solutions and FEM is done, showing good correspondence and a relatively good accuracy.

The model is then validated with a displacement input from the FE model with manually introduced noise values, in order to emulate a DIC response, showing a relatively low sensitivity to a normal distribution of noise.

Once the model is proven to behave as required, a real experiment (explained in detail in Appendix D.3) is conducted using a DCB specimen with pure bending moments applied to its ends, taking pictures in the deformed state to obtain the displacement field via the DIC method. The behaviour of the model agrees with the expectations, but the stress prediction is a rough estimate of reality due to the high amount of variables involved in the experiment.

Thus, the conclusion of the previous semester project [Viejo et al., 2019] is the proof that shows that with the developed methodology it is possible to characterize a cohesive law avoiding some of the drawbacks that the present methods have. However, the capabilities of the method need to be addressed and studied in-depth with the focus on real experimental conditions.

The flaws of the methodology have proven to be mainly related to the experimental procedure and the numerous variables involved, both during the delamination experiment and the post-processing of the data. Moreover, the main concern apart from the experimental procedure is related to the robustness of the inverse parameter identification. The problems that have been identified are listed below:

- **Experiment**
  1. **DIC photos**: the acquisition of images has been found as a problematic process. Even though a high volume of pictures had been taken during the delamination process, the number of images that were apt to be analysed was limited due to poor quality. Out of more than 50 images taken, 6 were providing accurate displacements using the DIC method and only 2 were finally evaluated for the results. This was identified to be caused by the high displacement rate of the machine during the specimen, that caused the delamination occur too fast to take good quality pictures.
  2. **Synchronization of equipment**: the experiment was performed with a setup that included a DIC setup, a moment rig machine and two inclinometers. Each piece of equipment was independent of each other (measurements were taken using a different device for each) and the synchronization of the results to correspond to a certain time during the delamination process was found to be troublesome. In this case, the moment value selected as input on the beam based model was not the one corresponding to the deformation state shown by the DIC analysis, which caused results to be incorrect. This situation caused an extra step in the post processing of the data
  3. **Moment machine**: Pure moment applied to the ends of the DCB produces a crack

that translates once it becomes fully developed, which made the interpretation of the results easier. However, this implies that a special machine, designed just for this purpose, has to be available in order to apply this methodology. The use of this machine involves the need of measuring a high number of variables with different equipment components. If the developed methodology aims to be relatively simple to apply, it must be able to be performed with more accessible equipment.
4. **3D effects study**: the methodology has proven to be effective according to 2D simulations, which are not an accurate representation of reality and cannot be related to represent the experiment accurately.
5. **Specimens used**: just one specimen was selected to be evaluated which is obviously not enough to describe the accuracy of the method.

- **Inverse Parameter**
  1. **Robustness**: The methodology has been shown to be highly sensitive to changes on the number of degrees of freedom used in the representation of the solution. Also, the position of these force points is shown to affect the solution quality drastically. Finally, it was demonstrated how the quality of the solution was even more sensitive to the already mentioned parameters when real data was used.

Some parts of the previous semester project [Viejo et al., 2019] have been omitted in this review, since only essential facts are highlighted to give an overview of the project.

1.4.2 Fourth Semester (Current)

As colophon to the previous semester's project [Viejo et al., 2019], an experiment was conducted and results that matched to a certain extent the guidelines introduced by the theory were obtained. However, it was clear that the weak point of the methodology was the experimental part. Thus, the scope on the fourth semester project is mainly focused on designing an experimental procedure that allows for a robust and accurate cohesive law characterization under real experimental conditions.

For a methodology to be effectively implemented, the correspondence to real experiments is an essential aspect. Therefore, a thorough evaluation of the effectiveness of the method under real experimental conditions must be done. This is achieved by the assessment of the main the possible sources of error that might affect the performance of the inverse parameter identification algorithm. All the variables (material, machine or DIC related) that are involved in the experimental setup are to be identified and quantified systematically, to evaluate the degree of variability of the cohesive law obtained, with the purpose of grading the level of confidence of the method.

Before the real experiments are conducted, a high fidelity 3D FE model is created to represent as accurately as possible the experimental conditions, in order to analyse and foresee *3D behaviours* that may affect the input data in an unexpected way.

The analysis of the variability of the cohesive law obtained is done using statistical analysis with the *Monte Carlo (MC)* approach, which helps to characterize the deviation of the resulting cohesive laws calculated combining all the quantified uncertainties. After the preliminary assessment, MC results are studied and correlation analyses are performed to estimate the parameter/s that most affect the error bounds of the final cohesive law. Thanks to this, modifications either in the inverse parameter identification model or in the experimental procedure are introduced in order to reduce the variability of the solution obtained.

Moreover, numerical methods are used to get an in-depth understanding of some of the variables defined by the user to characterize the cohesive law and to seek for a pattern that allows robustness of the solution. Then, changes in the *load function formulations* are applied accordingly and *constraints* to the solution are set, tailoring the system of equations to achieve a robust system that solves the issues encountered on the previous semester project [Viejo et al., 2019].

With all the information gathered from the preliminary studies, a final experimental procedure is designed and tested in the laboratory. This experimental procedure aims for an effective and relatively simple application of the methodology developed. Particular attention is paid to the difficulties (1 to 5 in the previous subsection) encountered during the previous semester project [Viejo et al., 2019].

From the previous description of the process to be followed in this semester, the objectives established for the present thesis are summarized as:

- Study the uncertainties and errors present in the measurements of the variables used for the calculations and how it affects the inverse parameter identification routine.
- Identify and study undesired high sensitivities of the inverse parameter model to the characterized uncertainties using statistical analysis based on the MC approach and the correlation studies between variables and solution.
- Improvement of the beam-based model to assure a robust and accurate behaviour of the inverse analysis. This is done with the aid of algebraic tools combined with the MC approach, that help to analyse the system and its behaviour depending on input parameters, in order to reduce the sensitivity of the system and improve the solution robustness. By reducing the sensitivity of the inverse parameter to the uncertainty of variables, the traction field predicted is uniquely related to the displacement field introduced.
- Develop a high fidelity 3D model using FE simulation with the aim of studying the effect of beam theory assumptions and 3D effects on the cohesive law variability.
- Design of an experimental procedure based on the results of the studies performed and previous experience.
- Validation of the method with the proposed experimental procedure under real experimental conditions using the MC approach for a set of different input displacements obtained from the DIC and the uncertainties for all the variables quantified.

This approach, if performed successfully, makes the methodology developed suitable for the evaluation of any type of composite material under mode I delamination in an effective and simple way.

Problem Formulation

The above discussion regarding the content of the present master thesis must be covered by the problem formulation stated in the following lines. The scope of the thesis can be formulated as: *Design of an experimental procedure based on a statistical study of the variability of a previously developed inverse parameter identification algorithm for the characterization of cohesive laws for composites under quasi-static delamination in mode I.*

1.5 Summary of the Method

From the work done in this master thesis, the main outcome is the validation of the developed inverse parameter identification tool against real experimental results. Input data is obtained from a experiment designed with the special purpose of reducing the influence of experimental uncertainties on the calculated solution. An overview of the tool is given in Figure 1.7. An example of the cohesive law characterized with this tool is shown in Figure 1.6.

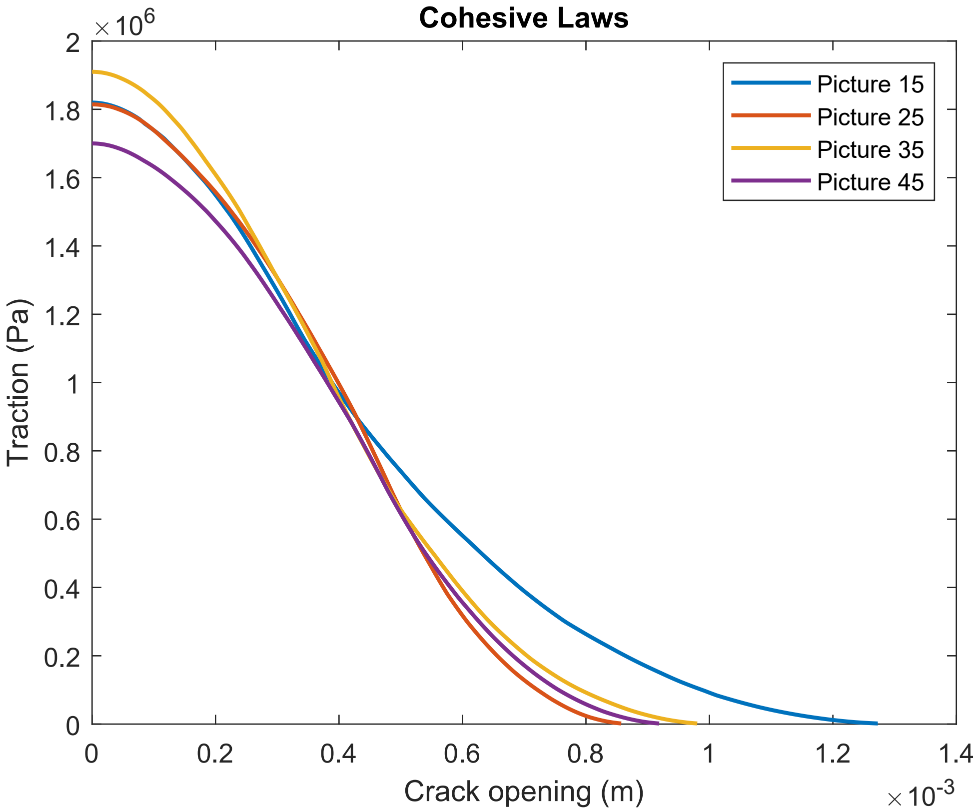


Figure 1.6: Example of cohesive laws obtained.

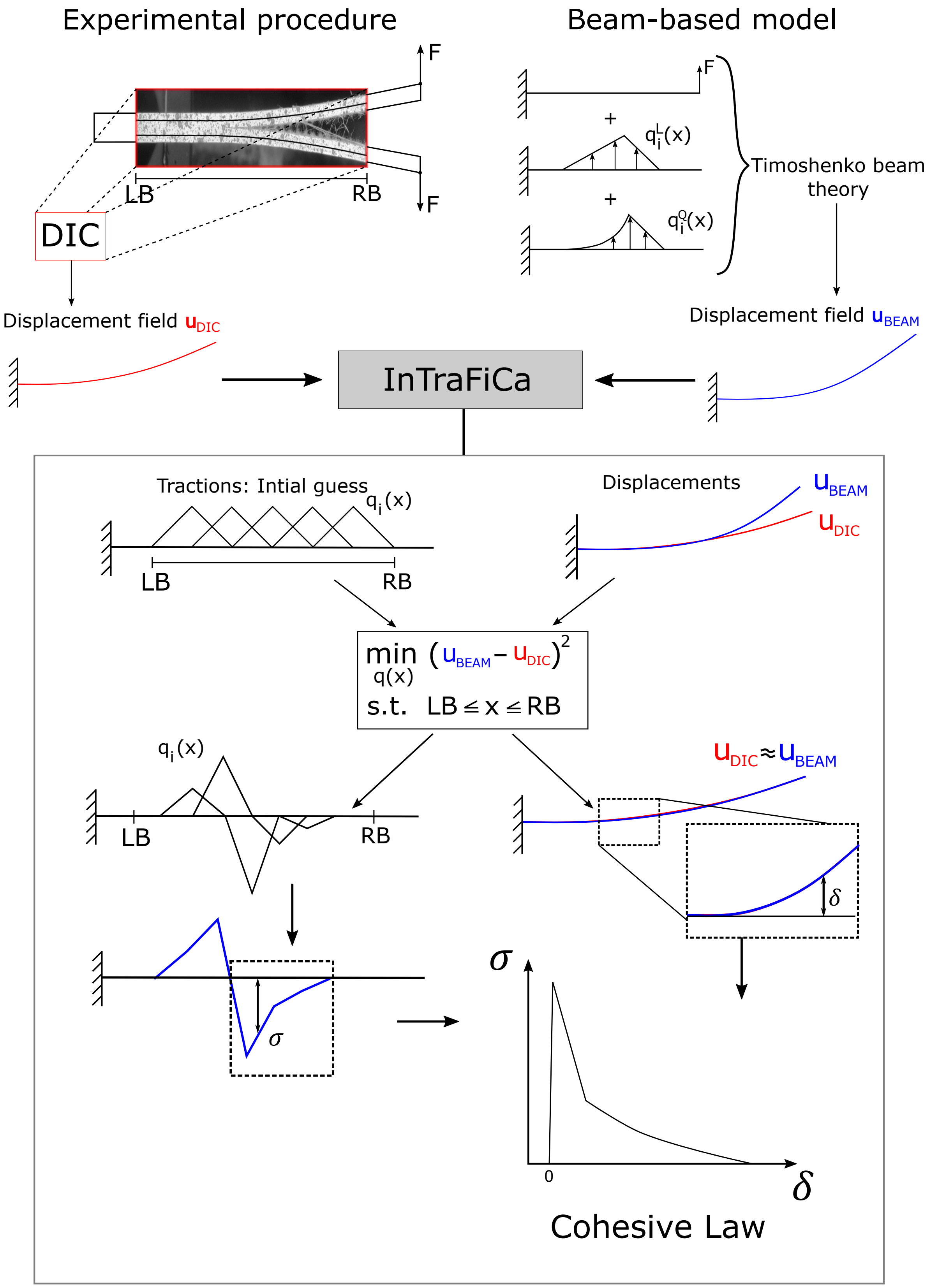


Figure 1.7: Sketch of the inverse parameter identification model of cohesive laws.

2 3D Model of the Proposed Experimental Procedure

In this chapter, the motivation for the creation of a FE-model that can reproduce real experimental conditions is introduced. Then, an study of the impact of 3D effects and misplacement of the hinges on the displacement fields used for the inverse parameter is done. Finally, the results of the study are analysed and discussed.

2.1 Requirements and Motivation for Building a 3D Model

For effectively simulating the experimental procedure, to aim for realistic results, a high-fidelity 3D FE-model of the DCB is developed. To make a model that can accurately represent the behaviour of the DCB during the test, an evaluation of objectives and possible inaccurate assumptions that can affect the results is made.

The main concerns of the high-fidelity model are to be able to:

- Capture the 3D effects
- Interpret imperfections present in real life
- Accurately describe the delamination process

2.1.1 Anticlastic Bending in 3D Beams under Delamination

This consideration is particularly important as a consequence of the method used for obtaining the displacement of the beam. The picture of the DIC is taken at the face of the DCB, assuming that the stresses and displacements do not vary along the width of the specimen. However, in the three-dimensional space, this assumption does not hold and it calls for a study of the influence of *anticlastic bending* on the beam.

Anticlastic bending is a phenomenon that occurs due to transverse bending of beams and plates, and it is caused by the Poisson effect. Using Theory of Elasticity for a beam in three dimensions (see Figure 2.1), the deformations due to pure bending in a beam causes the components of strain in the cross sectional plane to vary depending on the Poisson's ratio [Timoshenko and Goodier, 1951]. The lower part of the cross section compresses and the upper part expands, creating a *"nail-shaped"* beam.

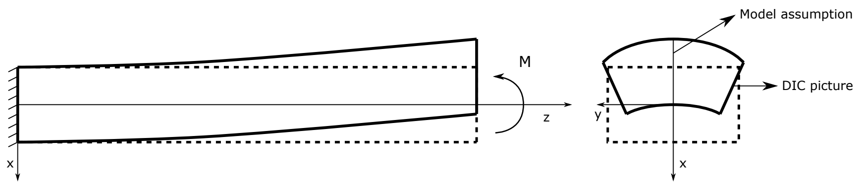


Figure 2.1: Anticlastic bending in a cantilever beam in a pure bending state.

This anticlastic bending affects the behaviour of the DCB specimen and the confidence of the results obtained from the DIC method. As seen in Figure 2.1, the image captured by the camera is of a face that is rotated with respect to the vertical, but is taken as if it is the mid-plane face when the inverse parameter model calculations are done. If a moment is applied to both arms of the DCB specimen, both upper and lower beam suffer from anticlastic bending while being delaminated (see Figure 2.2a).

This deformation of the cross section affects the stresses of the interface along the width of the cross section. At the outer faces of the specimen, the beams are pushing each other opposing the interface separation and delaying the delamination process. At the centre of the width, the anticlastic bending pushes the faces of the beam away, making the faces prone to delaminate before the edges of the beam. This effect causes the delamination front of a DCB specimen to be non-uniform across the width plane of the specimen (Figure 2.2b).

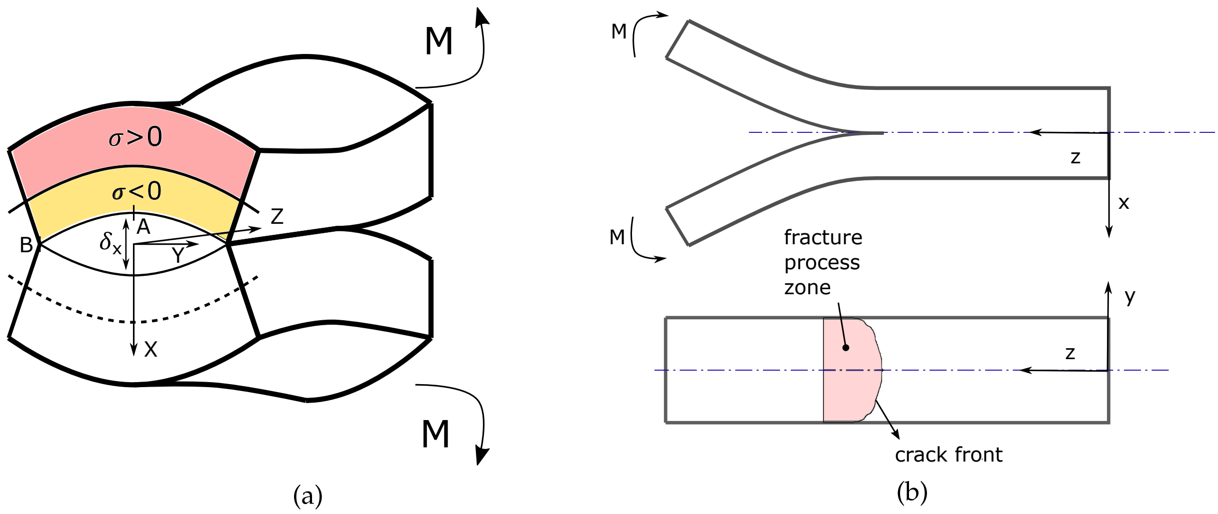


Figure 2.2: a) Anticlastic bending effect in a section of a Double Cantilever Beam in a pure bending state. b) Anticlastic bending effect on the evolution of the crack front across the interface.

Another aspect to be analysed in-depth in relation to the 3D model is the neutral line assumption. It is important to see whether the results of the beam-based model in Viejo et al. [2019] vary or not depending on the displacement field that is taken from the DIC.

2.1.2 Mispositioning of the Hinges prior to the Delamination Experiment

Particularly important are the areas where the force is being applied, in this case piano hinges are used for force transmission (see Figure 2.3).

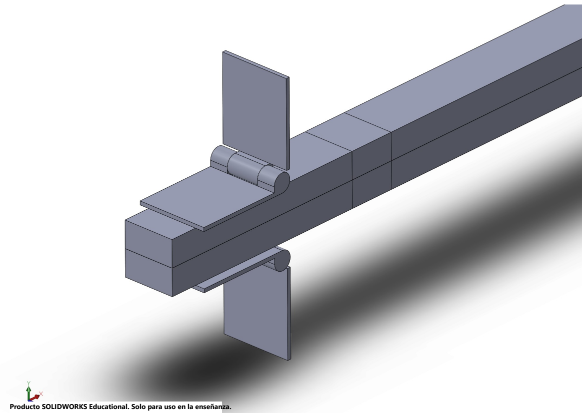


Figure 2.3: 3D CAD model of the DCB specimen with the piano hinges.

Some factors can affect the structure that might introduce errors on the displacement field used for the calculations e.g. forces are rotated with respect to the vertical axis (arrow in Figure 2.4). The method used to connect the hinges to the beam is by four bolts close to the edge where the crack is located. These bolts can easily produce a misalignment of the hinges due to the limited accuracy of the tools. Therefore, hinges are modelled in order to study the aforementioned situation and their effect on the results obtained.

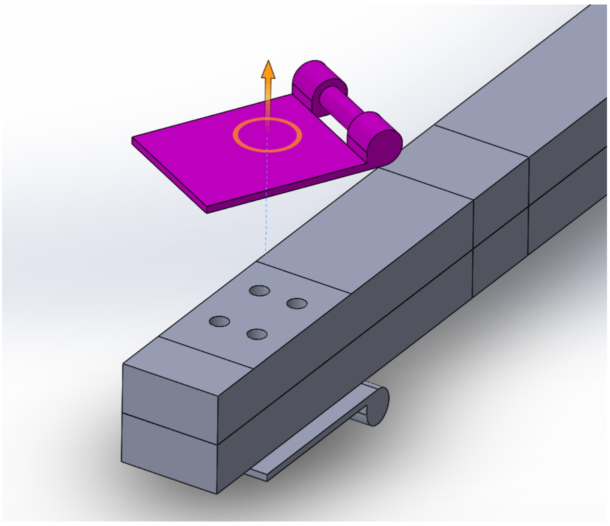


Figure 2.4: Exploded view of the rotated hinge face that can result from an incorrect hole alignment in the specimen.

Moreover, the fact that the deformation of the beam is an essential parameter for the inverse identification, makes important the consideration of support influences on the beam deformation. In this case, the hinges are made from steel and are stiffer than the composite used, therefore the deformation of the beam where the hinges are positioned is strongly affected by their presence, preventing the beam to deform freely.

2.1.3 Representation of the delamination process

The most essential feature of the model is that, apart from introducing the above characteristics, is able to accurately describe the delamination process under pure mode I. The model has to be able to represent correctly the stress state of the interface, making the parameters defined in the cohesive law to match the observed behaviour of the structure. For this reason, a suitable combination of boundary conditions, mesh characteristic and solver performance, that ensure an stable behaviour of the delamination process at all time steps, must be achieved.

2.2 Analysis of 3D-Effects on the FE-model Displacements

For a complete explanation on how the specimen is modelled in ANSYS Workbench, which features are used and the mesh convergence study of the model, see Appendix E.

2.2.1 Anticlastic Bending Effect on the Displacement

The vertical deformation along the width of the specimen is a matter of study, to confirm the theoretical considerations explained at the beginning of the present chapter.

The difference of vertical displacement between the lateral part of the beam and the central part of the beam, which is more significant at the delamination front, can be seen in Figure 2.5. Note that when the difference is negative, the position of the lateral face is lower than at the longitudinal symmetry plane of the DCB. This figure indicates that the maximum difference occurs near the crack tip.

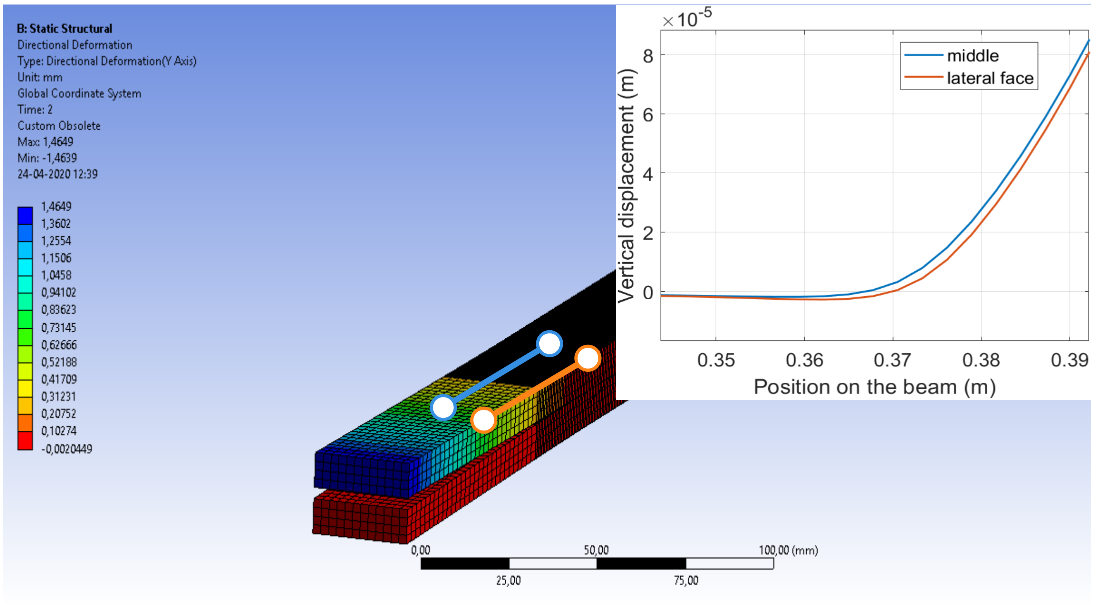


Figure 2.5: Vertical displacement differences between a path taken at the symmetry plane of the DCB and the displacement field at the lateral face.

The deformation just at the crack tip for the top surface and interface along the width of the specimen can be observed in Figure 2.6.

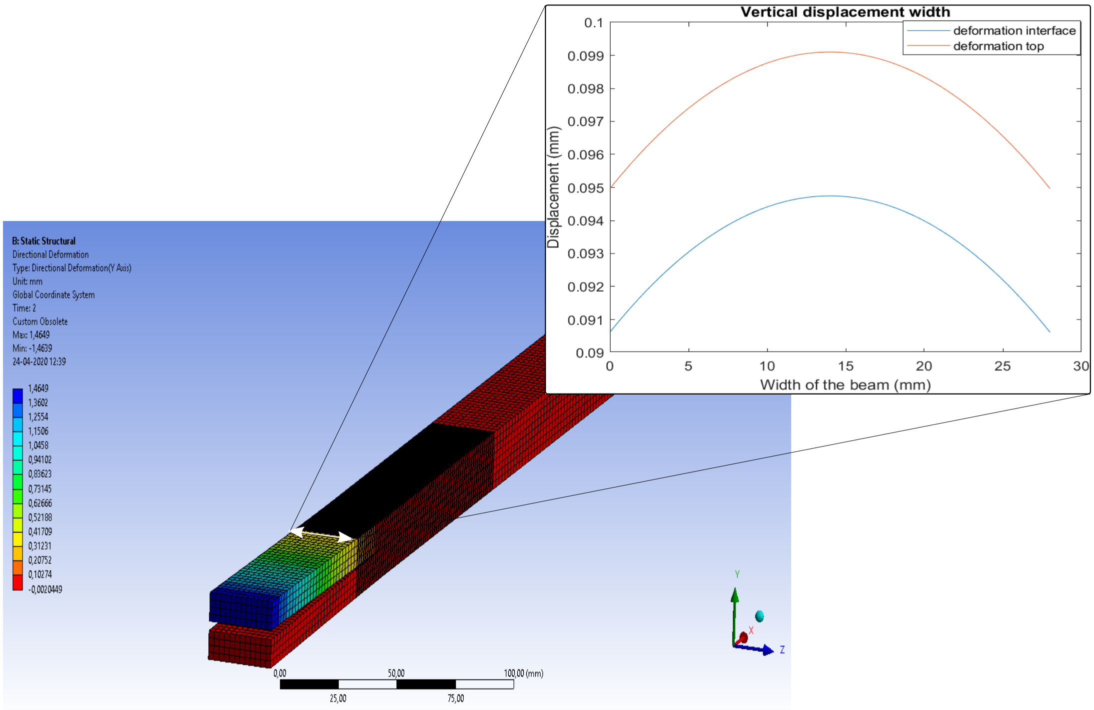


Figure 2.6: Difference in vertical deformation between lateral face and symmetry plane of the DCB.

In the x direction, the maximum rotation of the faces due to anticlastic bending correspond to the maximum difference in displacement. Thus, it is more accentuated close to the crack front, clearly seen in Figure 2.7.

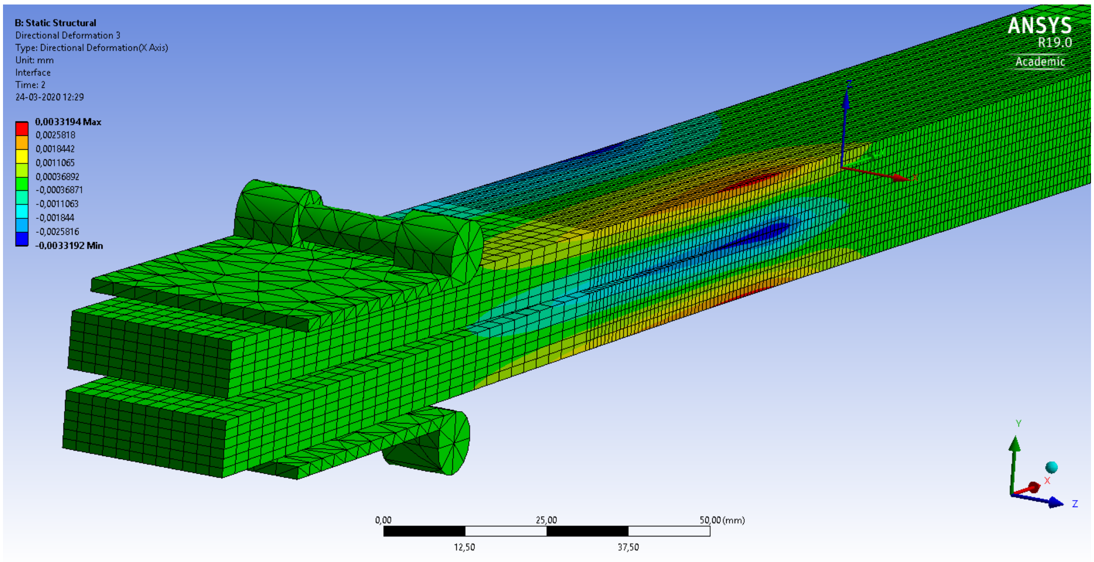


Figure 2.7: Deformation in the z direction during the delamination process.

The stress state observed at the interface of the DCB is not uniform along the width of the specimen (see Figure 2.8). It is seen how the maximum stress is reached at the edges of the specimen, while delamination at the centre has already occurred. This is caused by the anticlastic bending of the cross sections of the beams, that pushes the lateral faces against each other, making the stress needed for delamination to be reached after than at the centre.

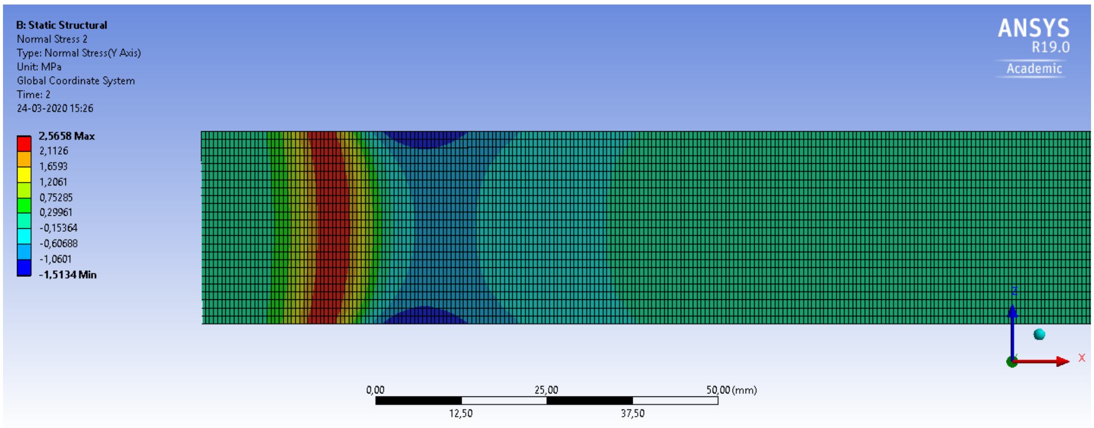


Figure 2.8: Normal stress in the solids at the interface.

The consideration of 3D effects is particularly important when the objective is to address the discrepancies between the inverse parameter model and experiment. The input of the inverse parameter algorithm is a displacement field obtained by means of DIC, with a picture taken to the lateral face of the beam. These lateral faces, in practice, suffer from the deformation caused by the anticlastic bending. However, the formulation of the inverse parameter identification algorithm is based on beam theory, which considers the cross section of the beam undeformable.

With the aid of the FE-model, the paths corresponding to the longitudinal symmetry plane can be extracted. This study aims to find out how the difference on displacement due to the anticlastic bending affects the solution process of the inverse parameter problem, and if the neutral fibre consideration holds even though the displacement field used as input is the one taken at the lateral face.

All the displacement fields considered for the present study are shown in Figure 2.9 and are defined below:

- Displacement fields available to the DIC method are located at the outer lateral faces of the beam. Two displacement fields are considered as input for this project, one is at the centre line of the face (path B in Figure 2.9) and the second one corresponds to the interface of the beam (path A in Figure 2.9).
- The displacement field of the neutral line of the beam (path D in Figure 2.9) is the path used to build the inverse parameter algorithm. This displacement field is the one used in the third semester project [Viejo et al., 2019] as input for the inverse identification algorithm and showed a good agreement with 2D simulations. The displacement field corresponding to the longitudinal symmetry plane at the delamination interface of the beam (path C in Figure 2.9). Once tractions are known, this displacement field represents the interface separation, used to build the cohesive law.

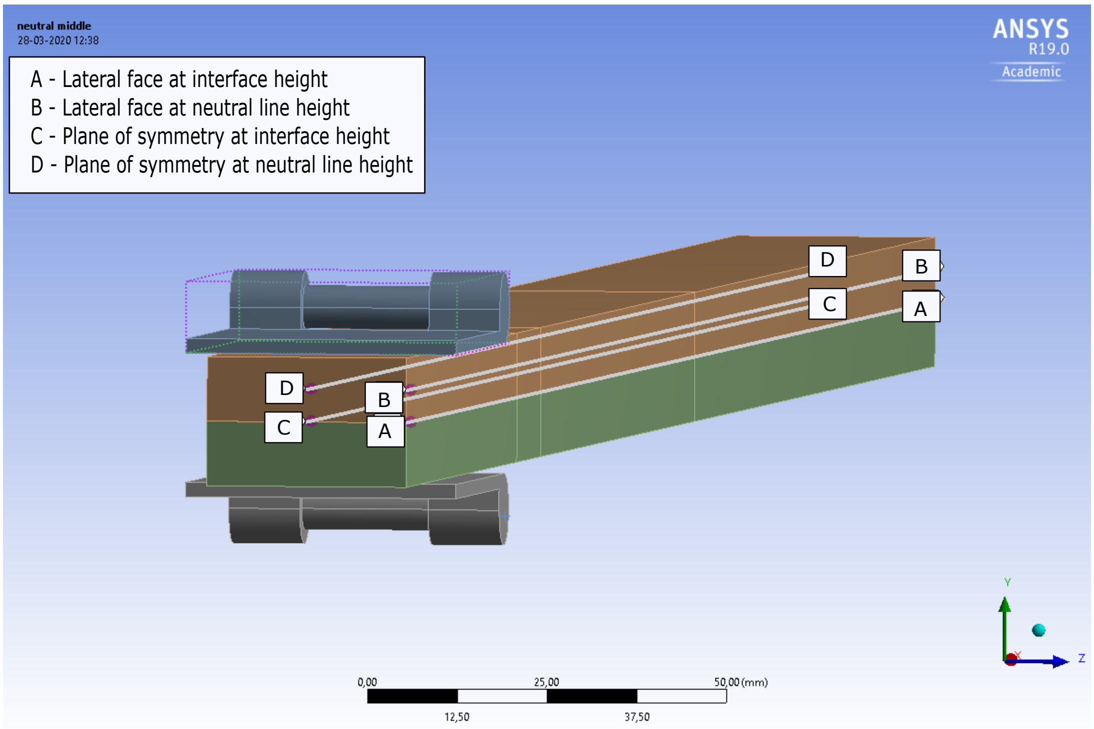


Figure 2.9: Displacement fields that are related to the DIC experiment and displacement fields used in the previous investigation.

Therefore, it is wanted to observe whether the the displacement fields taken at the lateral face under the effect of the anticlastic bending have a considerable impact on the results compared to the displacement fields used in the previous semester project [Viejo et al., 2019].

Discrepancies can be seen between the four paths selected, that can produce an erroneous result of the cohesive law calculated by the inverse parameter algorithm. It is observed how there is a clear shift in the paths located at the lateral face and at the longitudinal symmetry plane, specially at the crack tip where delamination is occurring. Moreover, paths at the interface suffer from a different deformation than the paths at the neutral plane, breaking the beam assumption of the underformable cross section. This finding advocates for the use of the paths at the neutral fibre as an input of the inverse parameter algorithm.

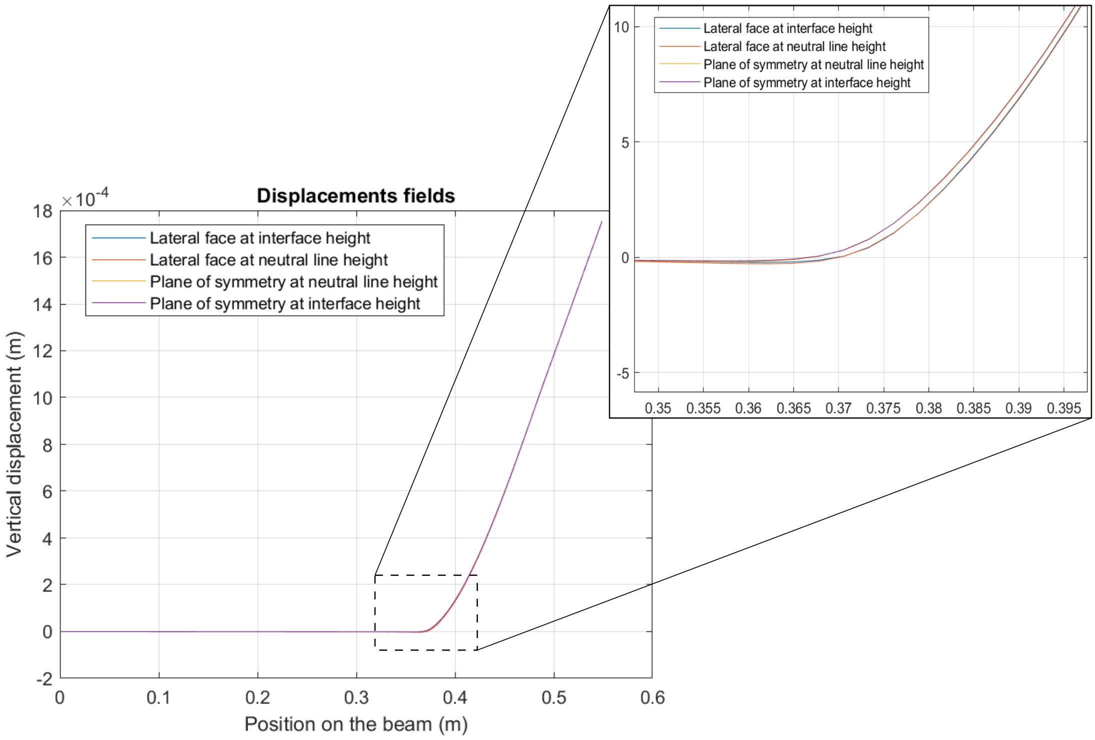


Figure 2.10: Vertical displacement of the top beam of the DCB specimen after the delamination with different paths taken at different areas of the geometry.

The impact of these deviations on the output of the inverse parameter identification is studied in the following paragraphs. The effect of these deviations in the inverse parameter algorithm traction estimation (using 12 degrees of freedom and positioning the tractions between x-coordinates $(0.2m, 0.45m)$) can be seen in Figure 2.11.

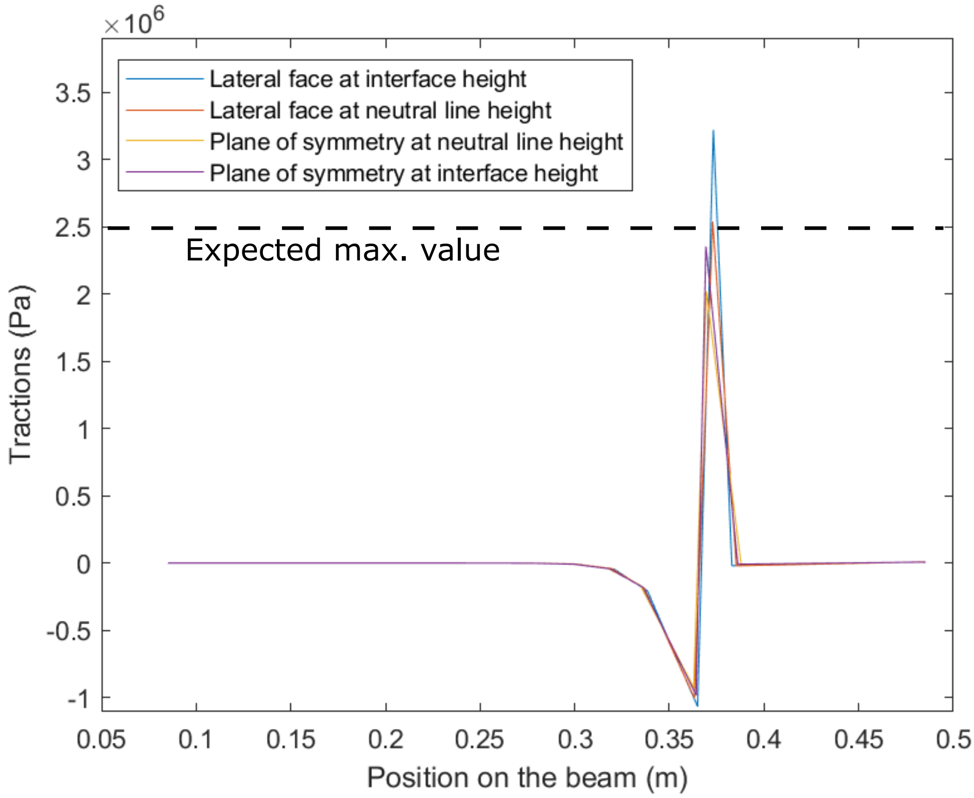


Figure 2.11: Stresses obtained when the different displacement fields shown in Figure 2.10 are used as input for the inverse parameter identification. These results are obtained using 15 degrees of freedom.

The stress predicted when the path on the lateral face at interface height is used as input overpredicts the maximum stress and gives a value around 3.3 MPa (see Figure 2.11). It has been mentioned in the previous subsection that the assumption of beam is broken due to the deformation of the cross section at the interface, so results with this displacement input are expected to be inaccurate.

The displacement field on the lateral face at neutral line height and the one at the plane of symmetry at interface height calculate almost the same value (around 2.5 MPa), predicting accurately the value established by the FE-model cohesive law (see Figure 2.12). Therefore, even though the outer face of the beam is affected by the anticlastic bending, its deformation resembles the behaviour of the interface displacement field at the longitudinal symmetry plane of the DCB.

The path of the plane of symmetry at neutral line height, on the other hand, underpredicts the maximum stress, being close to 2.1 MPa. The cohesive laws obtained for each of the paths and the cohesive law defined for the material in the FE software (corresponding to Figure E.1) are presented in Figure 2.12.

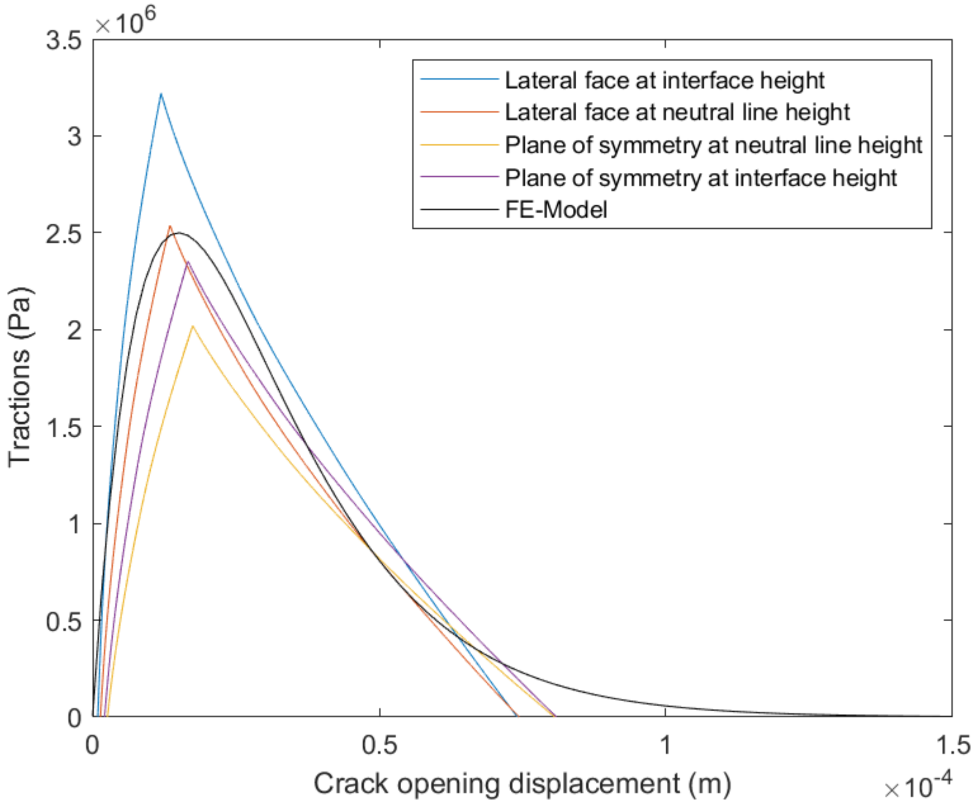


Figure 2.12: Cohesive laws obtained when the different stress fields shown in Figure 2.11 are input in the inverse parameter algorithm.

2.2.2 Hinges Rotation Effect on the Displacement

As suggested in Section 2.1.2, one of the effects that can cause a deviation in the displacement results captured with the DIC method can be a misplacement of the hinges that transmit the delamination force to the DCB specimen. The misplacement of the hinges is set from 0 to 2 degrees, when it is clearly seen by the naked eye that hinges are bad positioned. Note that the positioning of the hinges is not expected to be as bad as 2°, so this is a worst-case scenario.

With three FE-models, 5 simulations are computed, [-2° -1° 0° 1° 2°]. The displacement fields of each result set are analysed to see whether a deviation between them exists or not. The path that is going to be analysed is the one corresponding to the centerline of the face (path B in Figure 2.9) for all simulations. First of all, the displacements are shown in Figure 2.13. It is seen how the difference between the perfectly aligned hinges and the 1°/-1° rotation almost non-existent, whereas the difference between hinges rotated 2°/-2° is remarkable. In this case, observing the lower zoom image in Figure 2.13 the 2° line curves before the rest and shows the maximum vertical displacement of all the displacement fields taken for the study. Then, the -2° degree rotation is the one with the lowest vertical displacement, the to the irregular deformation of the beam.

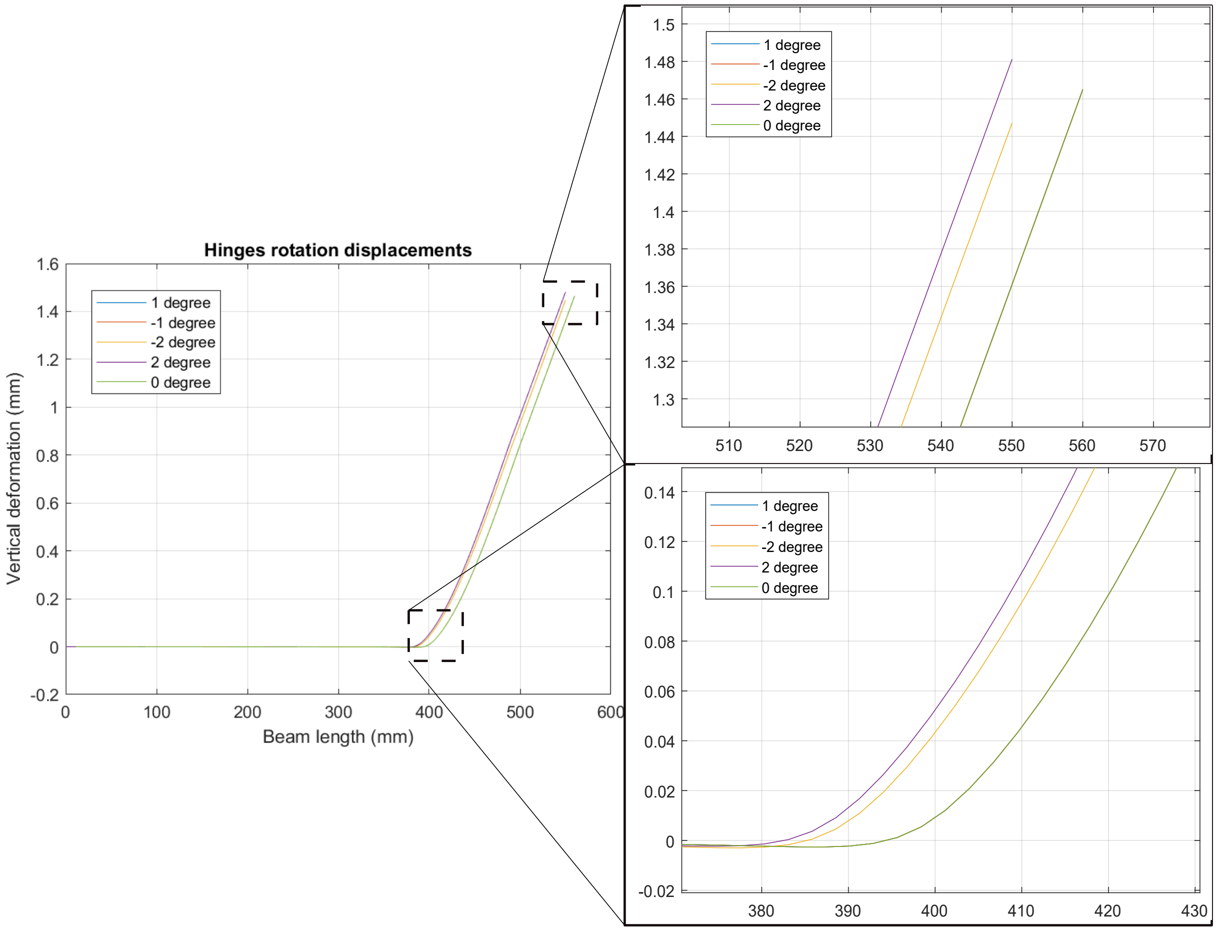


Figure 2.13: Different displacement fields generated by a misplacement of the hinges when mounting them on the DCB model.

The prediction of the cohesive stresses for each of the given displacement fields, with 15 degrees of freedom and the traction calculation limited between the interval $(0.25m, 0.45m)$, is shown in Figure 2.14. It is found that the magnitude of the cohesive stresses predicted is slightly different for the more rotated hinges (-2°/2°), having similar shape but lower onset traction than the rest of the cases. For a rotation of 1°/-1° the difference with respect to the perfectly aligned hinges result is negligible. It is known from the FE-model results that the maximum stress computed is around 2.55 MPa, which indicates that the rotation of the hinges under-predicts this value.

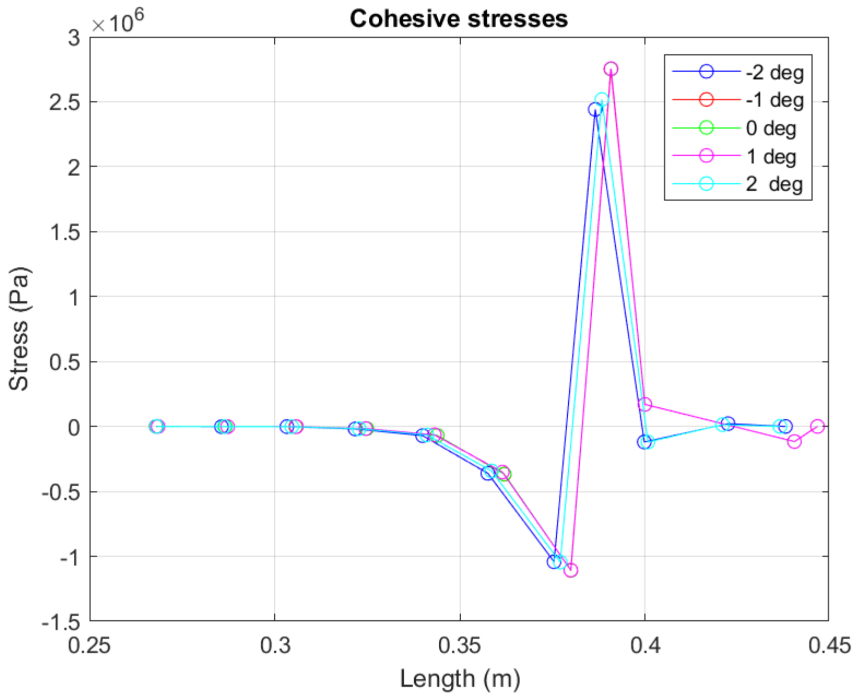


Figure 2.14: Cohesive stresses results extracted from the inverse parameter algorithm for the input displacements in Figure 2.13

The cohesive laws predicted are shown in Figure 2.15, with the FE-model cohesive law that represents the curve that should be imitated by the calculated solutions.

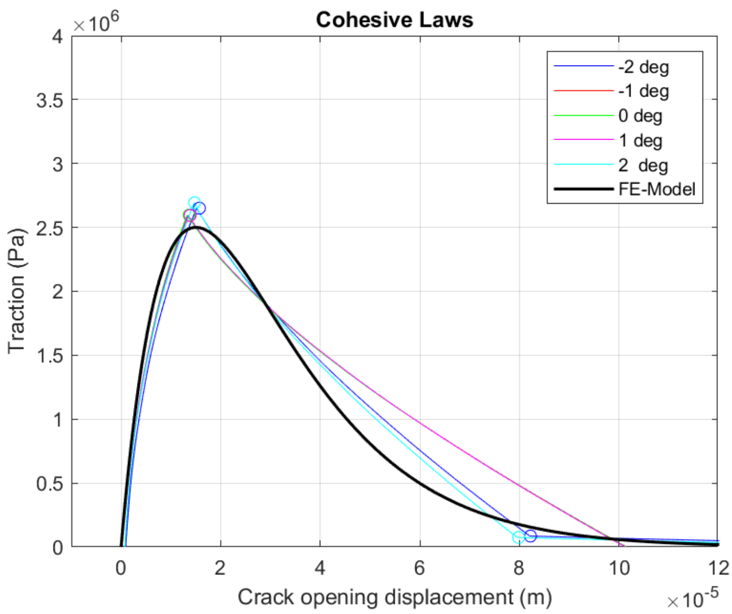


Figure 2.15: Cohesive law results extracted from the inverse parameter algorithm for the input displacements in Figure 2.13

It is shown how, the misplacement of the hinges produce a variation in displacement fields that causes the inverse parameter algorithm to predict a different onset traction. Moreover, it is observed that the last part of the cohesive law, corresponding to the fiber bridging area, is not accurately described when the rotation of the hinges increases.

2.2.3 Conclusions on the Hinges Rotation and Anticlastic Bending Effects on the Displacement

The conclusions for both studies regarding 3D considerations are presented now

- From the study of the anticlastic bending, and the effect on the lateral face displacements, it is clear that there is a slight difference on the displacement field values. However, even though the displacements are different, the displacement fields at the lateral face and at the longitudinal plane of symmetry do not cause a significant difference on the cohesive laws calculated.

  Therefore, the analysis shows how for this specimen with its particular properties, the effect of the anticlastic bending on the lateral faces does not suppose a big error on the computed solution. Moreover, it is also found that the displacement field to be used as input must be the one corresponding to the centerline of the beam at the lateral face (Path B in Figure 2.9).
- For the study of the effect of the hinges rotation on the cohesive law calculations, it can be seen how a big rotation makes accuracy of the solution obtained to be not good enough. The maximum traction seems to be closer to the expected one, but the shape of the cohesive law obtained at the bridging area does not represent the appropriate behaviour. However, these results are observed for a rotation of the hinges of 2^{o}, which has been proven to be easily detected by the naked eye. Therefore, the situation depicted should never occur in an experimental procedure, as the specimen should be labeled as not valid for the analysis.

3 Uncertainty Quantification: DCB Specimen and Experimental Procedure

This chapter is rooted in the theory presented in Appendix C, wherein the theoretical concepts of the methods to assess uncertainty are presented. The following paragraphs contain the characterization of the geometric data and the elastic properties of the DCB specimen which are of interest, together with their uncertainty values. The sources or error that are identified for the current experimental procedure are also characterized. The results from this chapter are later used in the sensitivity analyses of Monte Carlo in Chapter 4.

The present chapter is focused on giving the information of the data values which are introduced in the inverse parameter identification model (see Appendix D), together with their uncertainty. In order to obtain the value of the uncertainty of a quantity that is either measured or calculated, it is necessary to identify the primary sources of uncertainty that are present in the process of calculation or measurement. Stated in Appendix C, once these values are characterized, it is possible to accumulate them. Thus, it can be known how much the calculated quantity can deviate with respect to the expected value.

As it has been pointed in Chapter 1, the objective of the delamination experiment, which is to be carried out in the present semester, is to characterize the cohesive law of the tested specimens. The attainment of the cohesive law is the result of the information collected from a series of parameters, which have an uncertainty associated. This encourages a study of the sources of error and uncertainty present in the double cantilever beam experiment.

It is assumed that large errors are not present. Therefore, only systematic and random deviations are considered. The different uncertainty sources that have been found to be representative in the delamination experiment are separated into two groups: the test specimen and the experimental procedure. The first one comprises the geometric values and the elastic properties of the DCB specimen, while the second takes into account the errors and uncertainties associated to the devices that are used in the delamination experiment e.g. DIC camera or the tensile test machine.

The variability of the parameters is required in order to test the inverse parameter identification tool against the expected range of data that might be recorded during the real experiment. Therefore, from the study of the variability, the robustness of the inverse parameter identification routine can be assessed.

3.1 DCB Specimen Data

The specimens that are going to be tested have been cut manually from a whole plate. Thus, it is necessary to measure the dimensions properly. Additionally, there is the possibility that the material properties provided last semester do not match with the current specimen's ones. In composite materials, properties can vary noticeably from one plate to another. Therefore, they must be validated prior to the delamination experiment.

3.1.1 Geometric Data

The geometric parameters of interest are: the effective length of the specimen L, which is the distance from the hinge to the far-end of the specimen, the thickness h and the width w, since they are used in the Timoshenko beam formulation of the inverse parameter algorithm (see Appendix D). For each parameter, fifteen measurements are taken along the length of the beam. The length is measured with a millimetric ruler, while the thickness and the width are measured with a digital caliper. This indicates that each measurement is subjected to two different uncertainties; a type-A and a type-B. As presented in Appendix C, different treatments are given to assess these uncertainties. To start with, type-B evaluation is linked with the resolution of the device that is used. Following Equation (C.6) in Appendix C, the type-B uncertainty u_B can be obtained. For the type-A evaluation, the experimental mean and standard error for the fifteen measurements are calculated. Thus, the type-A uncertainty u_A is known.

Once these uncertainties are known, they can be accumulated following Equation (C.8) from Appendix C. The following table presents the geometric values with its associated absolute uncertainty. The mean values and uncertainties are all in millimeters.

	Length (L)	Width (w)	Thickness (h)
Mean	485	27.9	18.2
Type-A uncertainty u_A	0	0.011	0.018
Type-B uncertainty u_B	0.24	0.056	0.12
Combined uncertainty u	0.24	0.056	0.12

Table 3.1: Geometric values with its related uncertainty. Mean and uncertainty values are in millimeters.

3.1.2 Elastic Data

Due to the regulation of the access to Aalborg University caused by the COVID-19 measurements imposed by the Danish Parliament, the experimental characterization of the elastic properties explained in Appendix F could not be performed. Even though the group had access to the laboratory, it was decided to prioritize the delamination experiment over the elastic properties characterization. Therefore, the mean values of the elastic constants E and G, are taken from the previous semester project [Viejo et al., 2019], which are supposed to be a close approximation of the specimens used here. However, the available elastic constants of the specimens are nominal values instead of the values of the material, which may vary. The uncertainties of these values are taken from Muttashar et al. [2015], in which a similar material is characterized with the method explained Appendix F.

3.2 Sources of Error of the Experimental Procedure

The following paragraphs expose the sources of error and uncertainty present in the measurements taken during the delamination experiment of the DCB, which are mainly related to the DIC. It is believed that they might have an impact in the attainment of the interface tractions and the cohesive laws. These sources are stated in the following bullets.

- Noise of the DIC
- Misplacement of the hinges

The first bullet refers to the noise signal of the DIC, estimated via a Noise-Floor analysis [Jones and Iadicola, 2018]. This study was performed in the previous semester project [Viejo et al., 2019]. It consists in taking two pictures of the unloaded specimen to compare the displacement fields. The difference between the two sets of data is classified as the Noise-Floor. Figure 3.1, taken from the previous semester report Viejo et al. [2019], shows the difference in vertical displacement of the neutral line of the DCB between two consecutive pictures. It would be expected to obtain zero difference, as both pictures are taken in the exact same configuration. However, it can be seen how the displacement values differ from zero; thus, obtaining a pure noise field nf.

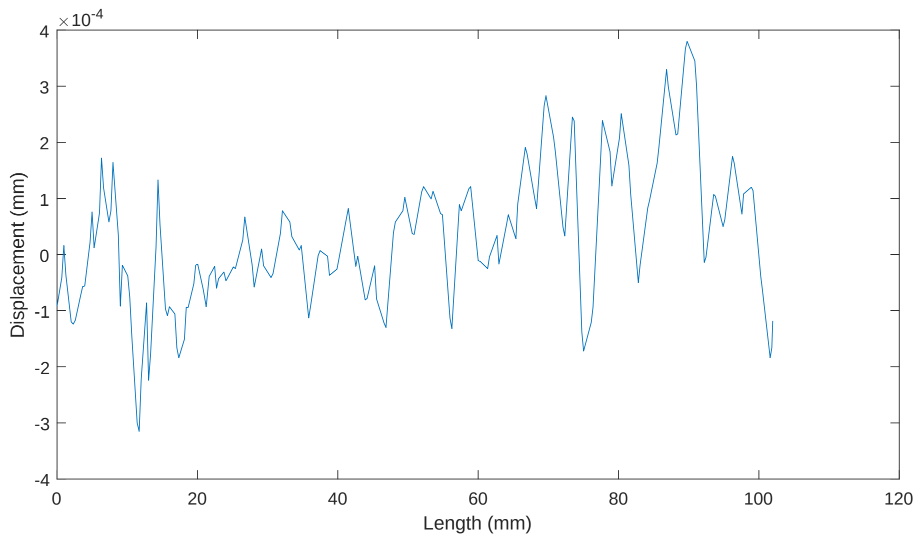


Figure 3.1: Noise-Floor analysis [Viejo et al., 2019].

As the noise is measured only once, a type-B evaluation is performed. Stated in Appendix C, when a type-B uncertainty is characterized, a statistical law must be chosen. It is assumed that the noise depicted in the above figure follows a normal distribution [Viejo et al., 2019]. In this case, the values of the mean $\overline{nf}$ and the standard deviation $s(nf)$, are stated in Equations 3.1 and (3.2). However, the value of the mean $\overline{nf}$ is intentionally set to zero, as it is desired for the noise to be unbiased.

$$\overline{nf} = 2.44 \cdot 10^{-8} \approx 0 \ \ m \tag{3.1}$$

$$s(nf) = 1.27 \cdot 10^{-7} \, m \tag{3.2}$$

With the value of the standard deviation, the absolute standard uncertainty of the noise field $u(nf)$, which is present in the DIC measurements, is known, see Equation (3.3).

$$u(nf) = 1.27 \cdot 10^{-7} \, mm \tag{3.3}$$

The second bullet takes into account the possible rotation of the hinges when they are bolted to the DCB specimen. This topic has already been covered in Chapter 2, where a thorough analysis of the 3D-effects is performed, and from where it is seen that the rotation of the hinges can influence the results of the inverse parameter identification model. For different rotations of the hinges, which are set to be ranging from -2º to 2º, the displacement fields obtained differ from the one where the hinge is not rotated. This behaviour is depicted in Figure 2.13, where the vertical displacement of the center line at the lateral face of the DCB specimen is plotted for the different orientations that are studied.

Unlike all the other quantities, which have been assessed in the present chapter, it is not possible to give a value of the uncertainty associated to the rotation of the hinges. Neither a type-A nor a type-B analysis can be executed as these results come from a numerical software. For each value of rotation, an entire finite element analysis must be solved. Therefore, it is believed that is unfeasible to generate a sample from where a statistical distribution can be approximated. Thus, this is seen as an error source rather than uncertainty. It is decided to characterize this error with a discrete uniform distribution in which the rotation values being [-2º,-1º,0º,1º,2º]. The displacement curve corresponding to each value of the rotation of the hinge is obtained from the FE-model, see Chapter 2.

4 | Analysis of the Inverse Identification using Monte Carlo

The Monte Carlo (MC) algorithm developed for assessing the sensitivity of the inverse parameter identification algorithm result to the uncertainty of the input data is explained in this chapter. First, the functioning of the algorithm itself is shown. Then, the performance of the inverse parameter identification routine built during the previous semester project Viejo et al. [2019] is evaluated with this algorithm. Correlation analyses are performed on the results to identify the parameters driving the error of the characterized cohesive law. Finally, the results obtained are used as motivation for the in-depth study of the mathematical tools in Chapter 5 and the redesign of the inverse parameter identification procedure in Chapter 6.

As explained in Section C.2 in Appendix C, MC methods can be used to introduce the randomness of real-life processes to a deterministic numerical model. This allows to obtain an interval in which the sought property is likely to lie when the uncertainty of the input is known. In this project, the purpose of the MC simulations is to obtain an estimate of the confidence interval of the calculated cohesive law when the input data are subjected to the uncertainties characterized in Chapter 3. Obtaining then, the sensitivity of an inverse parameter identification routine to the characterized uncertainties of the input. The displacement data used as input is generated using the 3D high-fidelity FE-model described in Appendix E, emulating an experimental DIC measurement.

A correlation analysis is carried out using the results of MC to determine the impact of each parameter on the results variation. A verification of these results is also done in order to validate the tool.

All the calculations are done in the program MATLAB. The scripts corresponding to the MC analysis are displayed in Appendix I.

4.1 MC Algorithm for Uncertainty Characterization of Cohesive Laws

4.1.1 Input Data of the Monte Carlo Algorithm

All the parameters used in the inverse parameter identification procedure have been characterized in terms of uncertainty in Chapter 3. The present MC algorithm consists of generating random values of these parameters, following the characterized distributions, and performing the inverse parameter identification of the cohesive law with these values. The input parameters are listed in Table 4.1 with their standard deviation if the distribution is normal, or the interval of the distribution for uniform distributions. The MATLAB functions *randn*, *rand* and *randi* are used to sample normal, continuous uniforms and discrete uniform distributions, respectively.

Parameter	Distribution	Mean	Standard Deviation	Interval
Flexural rigidity (EI)	Normal	78.16 Nm^2	3.44 Nm^2	-
Shear rigidity (kAG)	Normal	462000 N	46200 N	-
Length (L)	Normal	0.485 m	0.0014 m	-
Force (F)	Normal	$^*\mu^F$	0.5% of μ^F	-
Width (w)	Normal	0.028 m	5.9×10^{-5} m	-
DIC noise signal	Normal	0 m	$U([0,2.54\times10^{-7}])$ m	-
Hinges rotation	Discrete uniform	-	-	[-2º,2º]

Table 4.1: Distributions of the data used for the inverse parameter identification of cohesive laws. *The mean value of the force depends on the value of the rotation of the hinge

The symbol μ_F in Table 4.1 refers to the mean value of the force, which is slightly different for each value of the rotation. The mean values of the force are extracted from the FE-model for each of the rotation values of the hinges. They are shown in 4.2.

Hinges rotation	-2º	-1º	0º	1º	2º
Mean force	134.91 N	134.44 N	134.31 N	134.44 N	134.91 N

Table 4.2: Values of the force applied on the FE-model to provoke the delamination of the specimen. They are used as mean values of the force for each value of the rotation.

It must be emphasized that the standard deviation of the *DIC noise signal* in Table 4.1 is not a constant value. It is decided to change the noise signal in between samples of the MC algorithm to be able to apply correlation. If the same amount of noise is always introduced, it is impossible to measure the effect of the noise over the results. Therefore, the calculations of the cohesive law are done with different noise magnitudes added to the displacements to be able to evaluate the impact of this perturbation on the results. The *DIC noise signal* follows a normal distribution centered in zero with a standard deviation that changes in between samples, creating noise signals of different magnitudes. The standard deviation s of the noise is changed following the uniform distribution

$$s(\text{DIC noise}) = U([0, 2.54 \times 10^{-7}\, m]) \tag{4.1}$$

These values of the interval are chosen because the resulting average value of $s(\text{DIC noise})$, is $\bar{s}(\text{DIC noise}) = 1.27 \times 10^{-7}$ m, which corresponds to the experimental value displayed in Chapter 3.

4.1.2 Algorithm Description

One of the problems arising when trying to define a confidence interval for the cohesive law is that the cohesive law is a function and not a scalar value. As a result, the statistical tools explained in Appendix C have to be adapted. In the routine *InTraFiCa*, the resulting cohesive law is defined by a finite set of traction-separation values $[\sigma_j, \delta_j]$, not by a continuous function. To overcome this, the values of separation δ_j used to define a cohesive law are maintained throughout the different simulations, and the values of the traction σ_j corresponding to these separation values is what the routine calculates. After several calculations, a set of traction values σ_{ij} are available for each separation value δ_j. Doing this, the traditional statistical approach used to find the confidence interval of a scalar variable can be applied to the traction values σ_{ij} of each of the separation values δ_j. Therefore, a set of mean values $\overline{\sigma}_j$ and confidence intervals, defined by a lower bound $\sigma_{LB,j}$ and an upper bound $\sigma_{UB,j}$, are obtained for the cohesive law. See Figure 4.1.

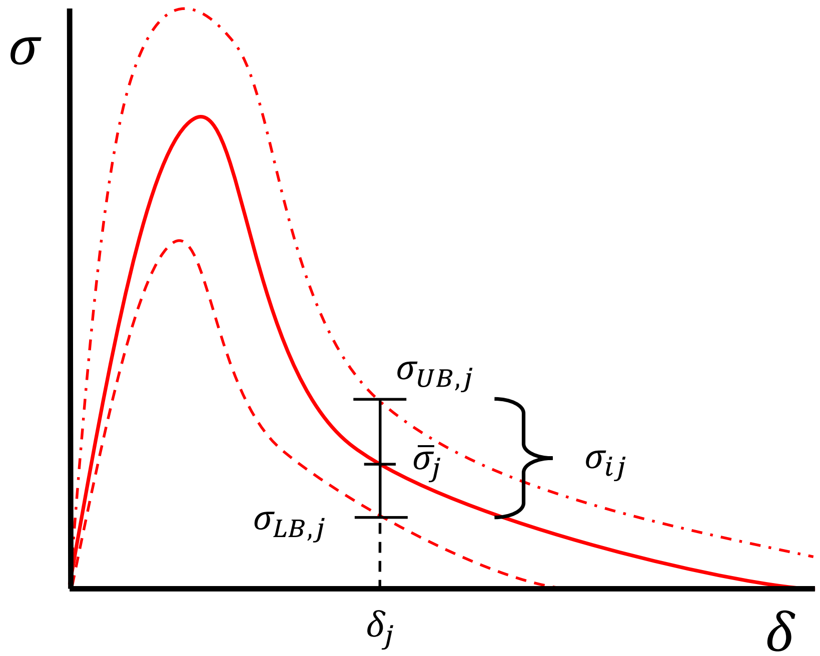


Figure 4.1: Sketch of the resulting mean cohesive law $[\delta_j, \overline{\sigma_j}]$ and upper $\sigma_{UB,j}$ and lower $\sigma_{LB,j}$ bounds.

The MC algorithm created in this project is sketched in the flowchart displayed in Figure 4.2. *Data of uncertainties* corresponds to the data in Table 4.1, which is used by the *Random sampling function* to generate a sample of random values R_{ik}. A random value for each of the $k = 1, \ldots, 7$ parameters in Table 4.1 is calculated following the displayed statistical distributions, where i corresponds to the number of the sample. This set of parameter values R_{ik}, is given to *InTraFiCa*, the inverse parameter identification routine, which returns a cohesive law.

For each value of separation δ_j, a value of traction is calculated for each sample σ_{ij}, where $j = 1, \ldots, N$, being N the total number of points describing the cohesive law. The value of N is chosen by the user together with the values of δ_j. Even though the separation values vary in between analysis, the traction values are interpolated at the specified separation values δ_j for all the analysis. To avoid losing information during the interpolation, $N = 10000$ is used in all the analyses. Moreover, the maximum separation value δ_N is set to be higher than any separation value with nonzero traction to capture the whole cohesive law.

After calculating the cohesive law corresponding to a number of samples M, the result is a matrix $\sigma_{ij} \in \mathbb{R}^{N\times M}$, having in each row the traction values corresponding to the cohesive law of the $i-th$ random sample R_{ik}. For each separation value δ_j, a mean traction value $\overline{\sigma_j}$ can be calculated for a sample size M as

$$\overline{\sigma_j}^M = \frac{1}{M}\sum_{i=1}^{M}\sigma_{ij} \tag{4.2}$$

The set of traction-separation values $[\overline{\sigma_j}, \delta_j]$ is termed *mean cohesive law*.

The lower and upper bounds of the confidence interval, $\sigma_{LB,j}$ and $\sigma_{UB,j}$, are defined for each separation value δ_j as

$$P(\sigma_{LB,j} < \sigma_{ij} < \sigma_{UB,j}) = \alpha \tag{4.3}$$

where α is the confidence level of the confidence interval and $P(a < x < b)$ means probability of x having a value in between a and b. They are calculated for each separation value δ_j as the two traction values which define an interval containing $\alpha \times M$ values of σ_{ij}, and leaving outside $(1-\alpha)\times M$ values. The area in between the curves defined by the set of values $\sigma_{LB,j}$ and $\sigma_{UB,j}$, which is an estimate of the area in which the cohesive law has an α probability of falling inside. Throughout the whole project $\alpha = 0.95$ has been used. For a sample size M, the bounds are defined as $\sigma^M_{LB,j}$ and $\sigma^M_{UB,j}$.

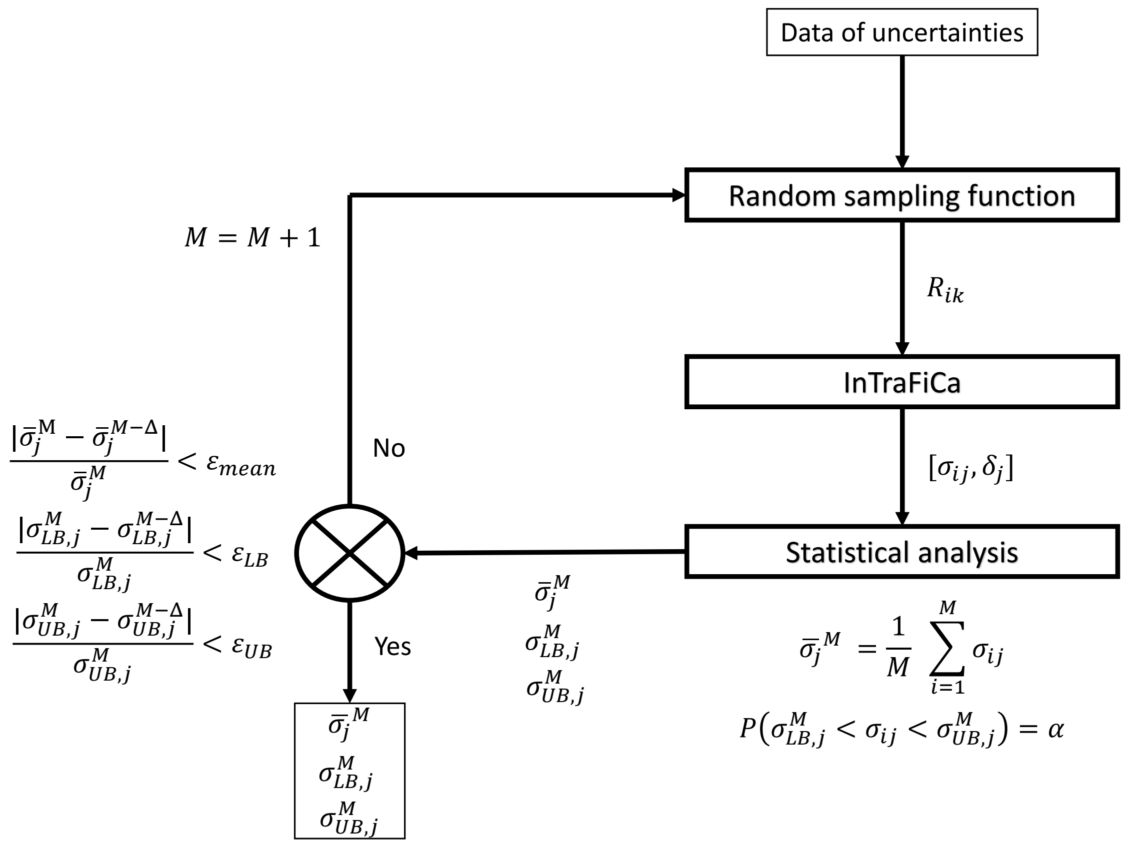


Figure 4.2: Flow chart of the MC algorithm.

As explained in Section C.2 in Appendix C, MC relies on the Law of Large Numbers. Therefore, as $i \to \infty$, the mean $\overline{\sigma}_j$ and confidence interval $[\sigma^M_{LB,j}, \sigma^M_{UB,j}]$ of the cohesive law approach the values of the population. For ensuring that these estimates are close to the population value, the stopping criteria is defined using the change in the variables of interest ($\overline{\sigma}_j^M$, $\sigma^M_{LB,j}$ and $\sigma^M_{UB,j}$) when increasing the amount of samples. For each separation value δ_j, the relative change in these variables is calculated when the sample size changes from $M - \Delta$ to M, where Δ is the amount of samples calculated in between two checks of the stopping criteria. This number is defined by the user and the only purpose of having $\Delta > 1$ is computational efficiency of the algorithm, because several inverse parameter identification calculations can be run in parallel increasing significantly the speed of the calculation. The parameters used for the stopping criteria are

$$\frac{|\overline{\sigma}_j^M - \overline{\sigma}_j^{M-\Delta}|}{\overline{\sigma}_j^M} \qquad \frac{|\sigma^M_{LB,j} - \sigma^{M-\Delta}_{LB,j}|}{\sigma^{M+\Delta}_{LB,j}} \qquad \frac{|\sigma^M_{UB,j} - \sigma^{M-\Delta}_{UB,j}|}{\sigma^M_{UB,j}} \tag{4.4}$$

A value of each of the ratios in Equation (4.4) is calculated for each separation value δ_j. These values are compared to the parameters ε_{mean}, ε_{LB} and ε_{UB} to decide whether the estimates are close to the population values. The value of these parameters is defined in Subsection 4.1.3. The MC simulation stops when the three ratios in Equation (4.4) are under ε_{mean}, ε_{LB} and ε_{UB}, respectively, for each separation value δ_j.

The reason of choosing these stopping criteria relies on the Law of Large Numbers. As the number of samples increases ($i \to \infty$), the parameters in Equation (4.4) tend to zero. Therefore, it is assumed that when these parameters are below a certain value (ε_{mean}, ε_{LB} and ε_{UB}), the approximation of the $\overline{\sigma}_j^M$, $\sigma^M_{LB,j}$ and $\sigma^M_{UB,j}$ can be considered a close approximation of the real mean and confidence interval of the cohesive law.

4.1.3 Stopping Criteria Study

The objective of this project is evaluating a methodology for determining the cohesive law, not determining it for a specific purpose. Therefore, no numerical requirement can be set on the result. In relation to the stopping criteria, there is no manner of setting a value for ε_{mean}, ε_{LB} and ε_{UB}, so it is decided to justify its value with a simple visual inspection of the obtained values of $\overline{\sigma}_j^M$, $\sigma^M_{LB,j}$ and $\sigma^M_{UB,j}$. When they do not seem to change after increasing the sample size, the MC simulation is considered completed. They all are set to 1%.

As an example, the progression of the parameters used for the stopping criteria is shown in Figure 4.3 when 10 traction d.o.f. are used in the inverse parameter identification and 95% as confidence level of the confidence interval. The values of the maximum relative changes in mean and bounds are displayed for each 200 new samples ($\Delta = 200$).

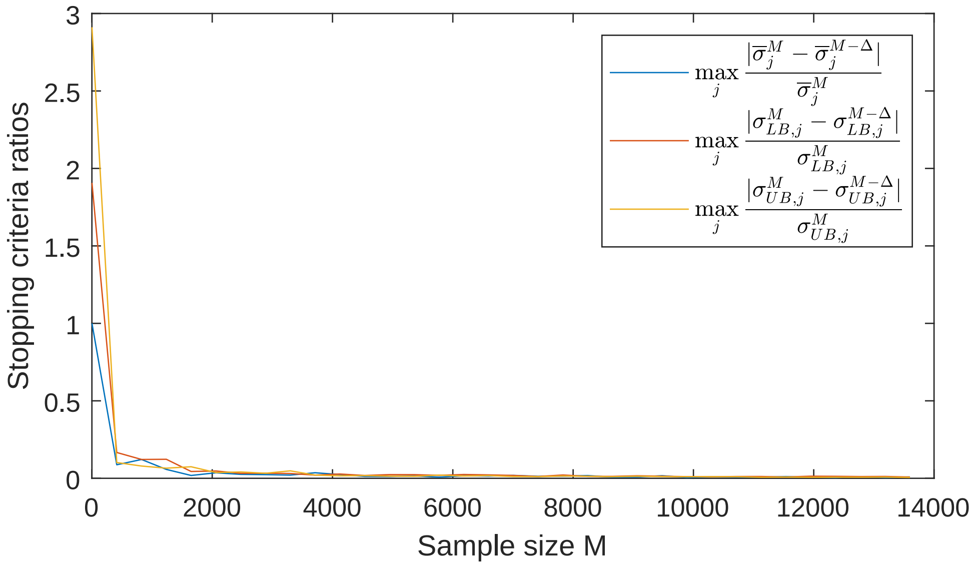


Figure 4.3: Progression of the ratios of the stopping criteria (Equation (4.4)) throughout the MC simulation using 10 traction d.o.f. on the inverse parameter identification routine. The parameters are calculated each 200 new samples. The displayed value of the parameters is the maximum for all the separation values δ_j

It can be seen how the values of the three parameters tend asymptotically to zero, as it is assumed to happen. When all the ratios in Equation (4.4) are below ε_{mean}, ε_{LB} and ε_{UB}, for each opening value δ_j, the simulation stops. The value of the stopping tolerance, 0.01, can be justified by displaying the calculated mean cohesive law $\overline{\sigma_j}^M$ and bounds $\sigma_{LB,j}^M$, $\sigma_{UB,j}^M$ for different sample sizes, displayed in Figures 4.4, 4.5 and 4.6.

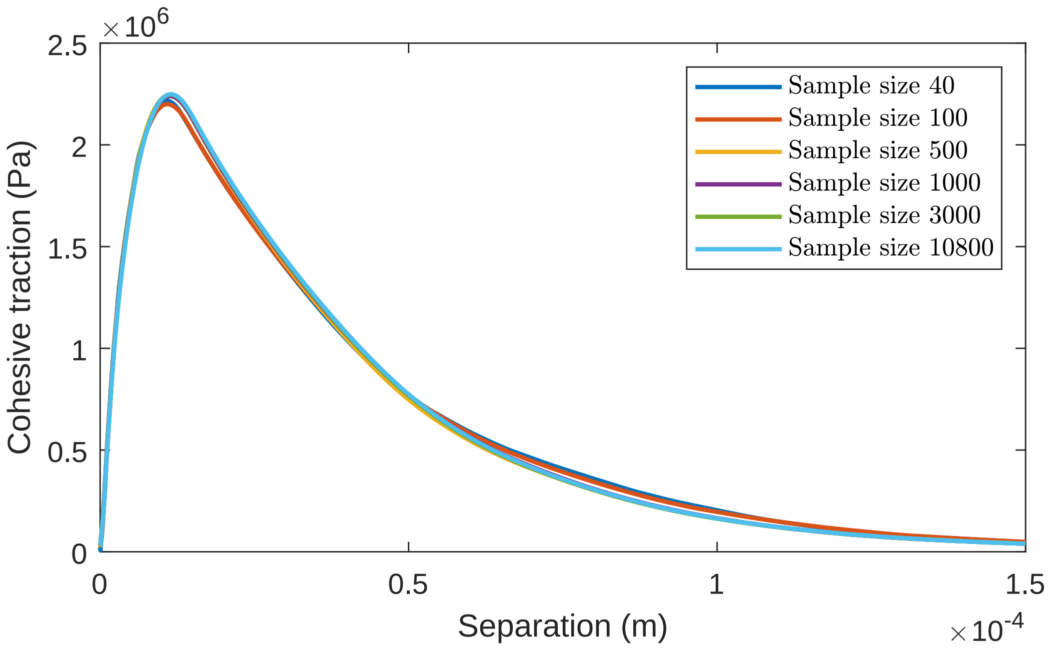


Figure 4.4: Change in the mean cohesive law $\overline{\sigma}_j^M$ throughout the MC simulation when 10 traction d.o.f. are used in the inverse parameter procedure. The legend shows the sample size.

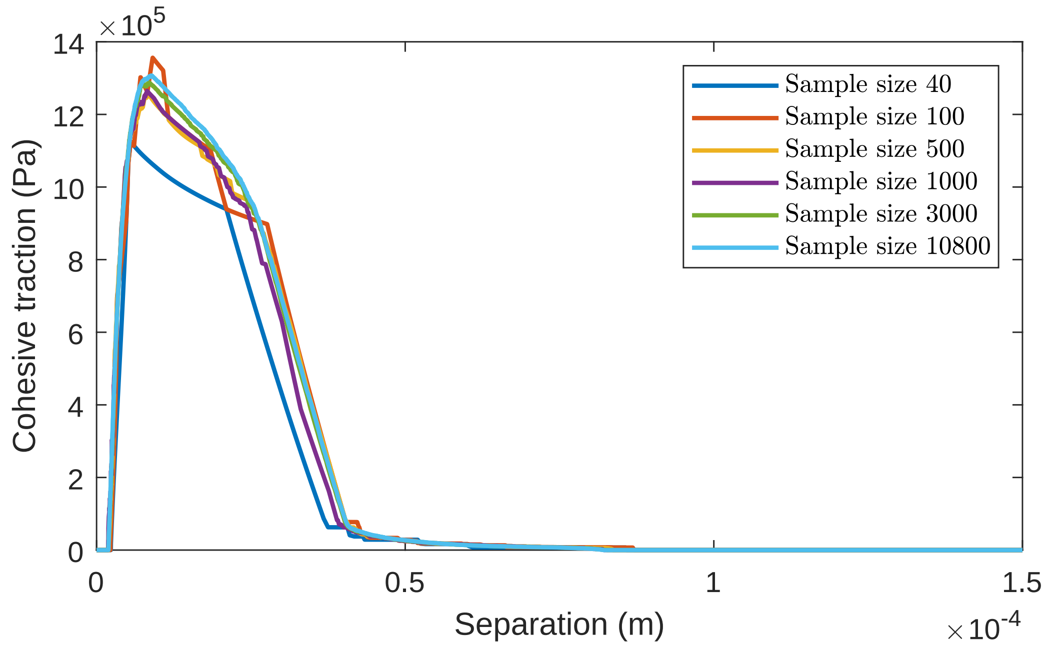


Figure 4.5: Change in the lower bound of the confidence interval $\sigma^M_{LB,j}$, for a confidence level $\alpha = 0.95$, of the cohesive law throughout the MC simulation when 10 traction d.o.f. are used in the inverse parameter procedure. The legend shows the sample size.

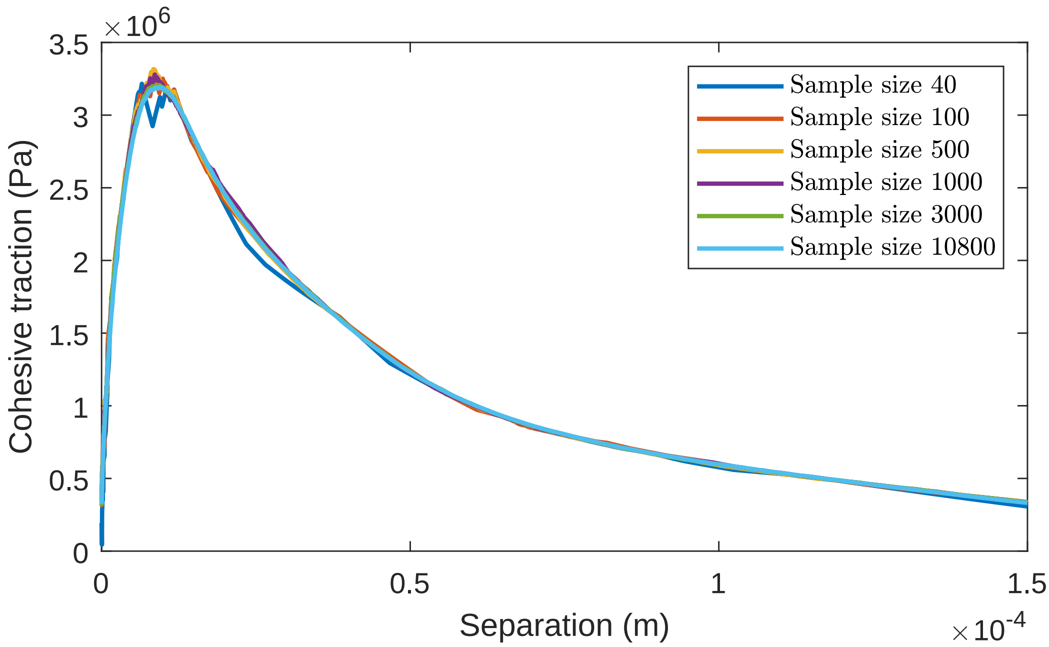


Figure 4.6: Change in the upper bound of the confidence interval $\sigma^M_{UB,j}$, for a confidence level $\alpha = 0.95$, of the cohesive law throughout the MC simulation when 10 traction d.o.f. are used in the inverse parameter procedure. The legend shows the sample size.

It can be seen how the curves of 3000 samples and 10800 samples are almost overlapping, which suggests the idea that a tolerance value of 0.01 is quite conservative. The same asymptotic behaviour is observed for all the analysis performed in this project, where no appreciable change in the curves is seen after a sample size considerably smaller than the final one. So the value of 0.01 is assumed to be conservative, however, it is decided to maintain it to ensure a representative sample size.

4.2 Analysis of the Third Semester Version of *InTraFiCa*

The 3rd semester version of *InTraFiCa* corresponds to the inverse parameter identification built in Viejo et al. [2019] and explained in Appendix D.

This simulation is run for different number of degrees of freedom of the calculated tractions. The initial guess provided to *fmincon*, regarding the position of the traction peaks, is equally spaced points along the whole beam. The results are displayed in Figures 4.7 and 4.8, where the continuous line corresponds to the mean cohesive law $\overline{\sigma}_j^M$ and the dashed lines to the lower bound $\sigma_{LB,j}^M$ and the dash-dotted line to the upper bound $\sigma_{UB,j}^M$ of the confidence interval at 95% of confidence level.

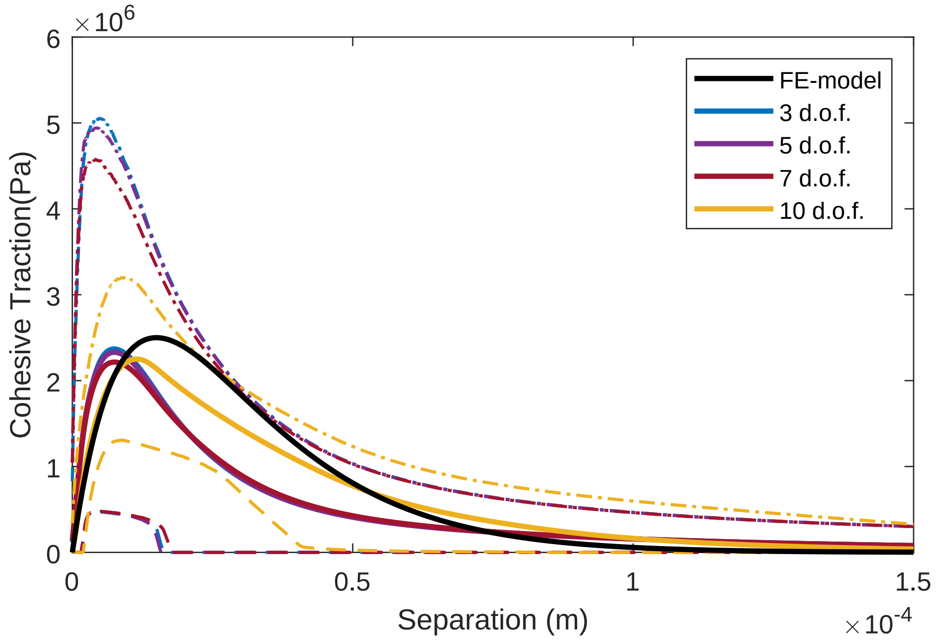


Figure 4.7: Resulting mean cohesive laws $[\overline{\sigma}_j^M, \delta_j]$ (continuous line) and confidence interval, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{UB,j}^M]$ (dash-dotted line), obtained from the MC algorithm using 3, 5, 7 and 10 traction d.o.f. The legend shows the traction d.o.f. used to calculated each of the curves and the curve labeled as *FE-model* is the one introduced as material property in the Finite Element model.

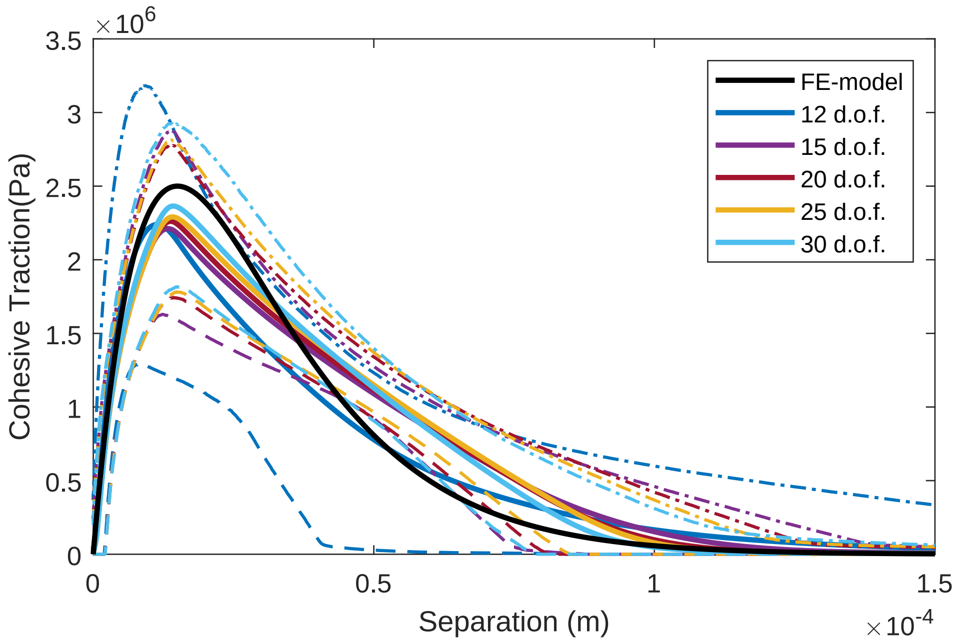


Figure 4.8: Resulting mean cohesive laws $[\overline{\sigma}_j^M, \delta_j]$ (continuous line) and confidence interval, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{UB,j}^M]$ (dash-dotted line), obtained from the MC algorithm using 12, 15, 20, 25 and 30 traction d.o.f.. The legend shows the traction d.o.f. used to calculated each of the curves and the curve labeled as *FE-model* is the one introduced as material property in the Finite Element model.

It is known from the previous semester project Viejo et al. [2019] that if the initial guess of the traction peak positions is distributed along the whole beam, most of the traction d.o.f. are "wasted" in calculating zero tractions, even though the nonlinear optimizer moves some of the traction peaks to the area where the interfacial tractions are nonzero. In order to check the capabilities of the system, a new series of analyses is performed in which the initial guess of the position of the traction peaks is equally distributed along the area where the tractions at the interface are nonzero. This area is known, as it can be seen in the FE-model postprocessing. Thus, the number of degrees of freedom used to calculate the tractions where the damage is occurring is increased. The results of the MC simulation are shown in Figure 4.9.

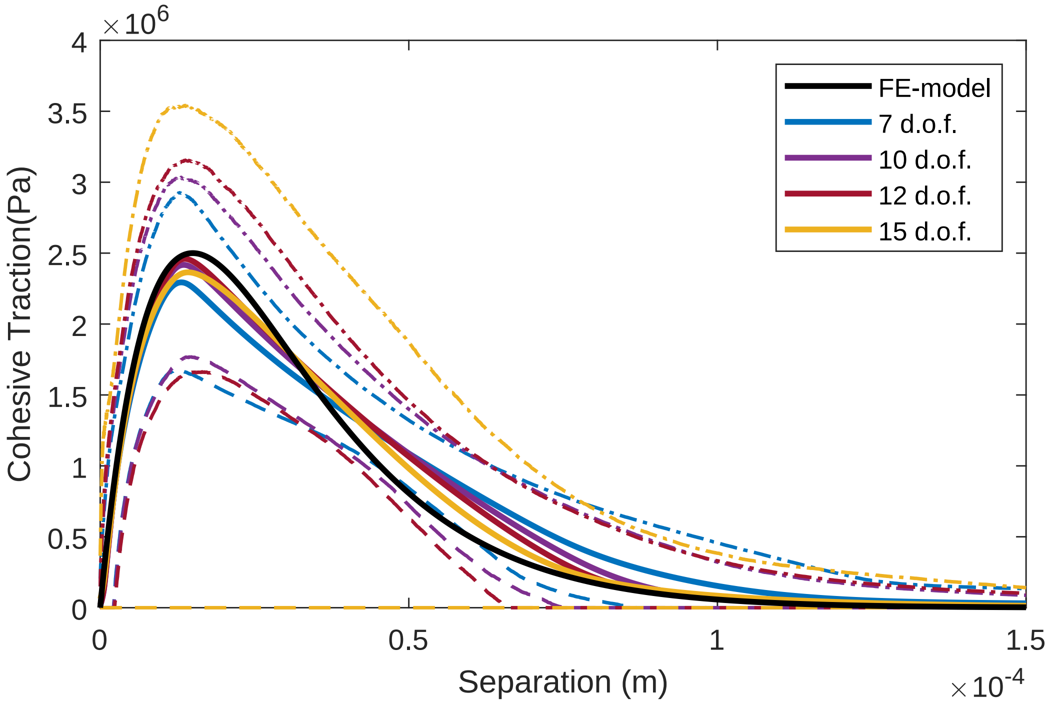


Figure 4.9: Resulting mean cohesive laws $[\overline{\sigma}_j^M, \delta_j]$ (continuous line) and confidence interval, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{UB,j}^M]$ (dashed-dot line), obtained from the MC algorithm using 7, 10, 12 and 15 traction d.o.f.. The initial guess of the traction peaks positions is equally spaced along the area of nonzero tractions at the interface. The legend shows the traction d.o.f. used to calculated each of the curves and the curve labeled as *FE-model* is the one introduced as material property in the Finite Element model.

It can be seen how, as the number of traction d.o.f. of the inverse parameter identification increases, the mean cohesive law tends to the FE-model cohesive law, see Figure 4.7. But after a specific number of loads, 15 in this case, the mean cohesive law barely changes, see Figure 4.8. Besides, when the number of traction d.o.f. over the area of interest, the area of the cohesive tractions, is increased, the confidence interval gets wider, as shown in Figure 4.9.

4.2.1 Correlation Analysis on the Monte Carlo Results

The correlation analyses are a tool that allows to estimate dependencies in between random variables, as explained in Section C.2 in Appendix C. In this project, correlation is used to evaluate how sensitive is the inverse parameter identification procedure to each of the input parameters R_{ik}. Thus, allowing to find the parameters driving the error for each analysis. In addition, correlation analyses can be used to prove the independence of the input parameters, ensuring that the routine works as intended.

To check the dependency of the input parameters in the random generation, the value of each parameter R_{ik} for each sample i is correlated with all the other parameters in pairs. This is done as a verification test of the code done in MATLAB. Tables 4.3 and 4.4 correspond to the

MC analyses with 10 traction d.o.f. The resulting *Pearson Product-Moment Correlation Coefficient* (PPMCC) is displayed in Table 4.3.

	EI	kAG	L	F	w	DIC noise standard deviation	Hinges rotation
EI	1.000						
kAG	-0.009	1.000					
L	-0.013	0.016	1.000				
F	0.007	-0.008	0.003	1.000			
w	0.005	0.006	-0.001	-0.001	1.000		
DIC noise standard deviation	-0.001	0.001	0.008	0.006	0.001	1.000	
Hinges rotation	-0.001	0.004	-0.011	0.003	-0.006	-0.007	1.000

Table 4.3: PPMCC in between each pair of input parameters. For each cell, the parameters correlated are the name of the column and the name of the row. The table values are symmetric w.r.t. the diagonal.

All the values of correlation coefficient, except when a variable is correlated with itself, are very close to zero, which supports the idea of true random number generation. The analyses are repeated with the *Distance Correlation* coefficient to capture any nonlinear correlation, see Table 4.4.

	EI	kAG	L	F	w	DIC noise standard deviation	Hinges rotation
EI	1.000						
kAG	0.014	1.000					
L	0.016	0.017	1.000				
F	0.015	0.017	0.013	1.000			
w	0.012	0.014	0.013	0.011	1.000		
DIC noise standard deviation	0.012	0.009	0.014	0.017	0.01	1.000	
Hinges rotation	0.012	0.013	0.015	0.174	0.012	0.010	1.000

Table 4.4: Distance Correlation coefficient in between each pair of input parameters. For each cell, the parameters correlated are the name of the column and the name of the row. The table values are symmetric w.r.t. the diagonal.

The results in Tables 4.3 and 4.4 show the same result: the variables are not correlated, which supports, again, the idea of true random number generation. However, in Table 4.4, the correlation coefficient value in between the force *F* and the *Hinges rotation* is substantially higher than any of the other values. This is caused by the dependence of the mean value of the force with the rotation, as explained at the beginning of this chapter. The mean value of the forces is different for the different rotation values, and the random numbers are generated accordingly. This relation is nonlinear, as it can be seen in Table 4.2, which is the reason why PPMCC does not capture this relation.

After performing the MC simulation, a set of traction-separation values $[\sigma_{ij}, \delta_j]$ are obtained for each set of parameters R_{ik}. To perform the correlation analyses, two variables must be correlated. In this case, the result consists on a set of values. Therefore, it is decided to perform a correlation analysis of each of the input parameters R_{ik} with respect to the calculated traction values σ_{ij} for each of separation value δ_j. The result is a set of correlation values for each separation value δ_j and each of the k input parameters. Instead of a correlation value, a *correlation curve* is obtained. The *Distance Correlation* coefficient is only calculated for certain separation values because, as said in Section C.2 in Appendix C, its calculation is computationally expensive. See Figures 4.10 and 4.11. As an example, the calculation of a single value of PPCMM in Table 4.3 takes around 0.001 seconds and a value of Distance Correlation in Table 4.4 takes around 10 seconds. These numbers are of course strongly linked to the sample size.

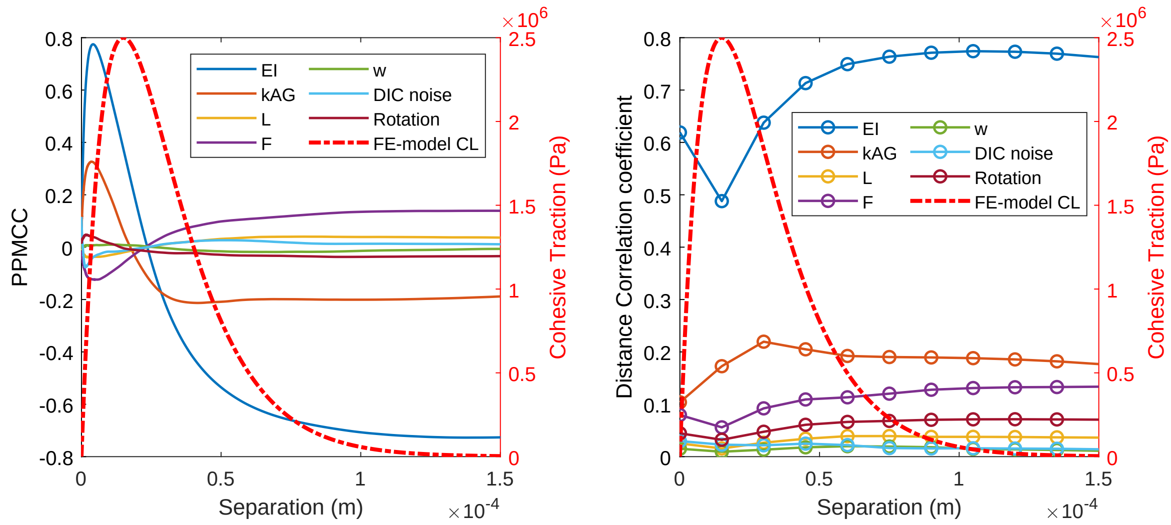


Figure 4.10: Correlation curve for each separation value when 3 traction d.o.f. are used in the inverse parameter identification routine. In the Distance Correlation graph, the points marked with a circle are the calculated points. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

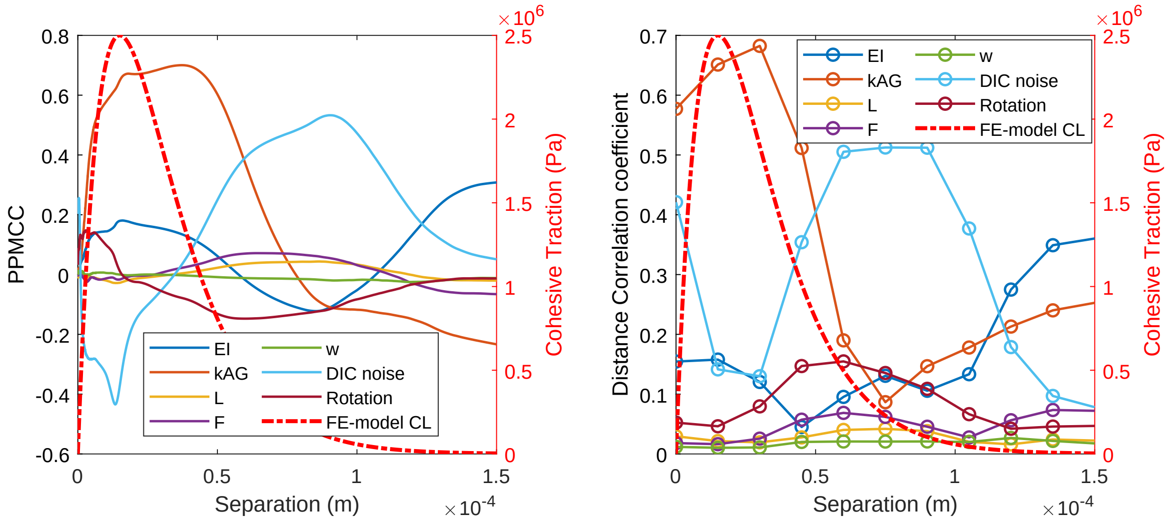


Figure 4.11: Correlation curve for each separation value when 25 traction d.o.f. are used in the inverse parameter identification routine. In the Distance Correlation graph, the points marked with a circle are the calculated points. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

It can be seen that both coefficients agree. The results from the correlation analyses show that, for a low number of traction d.o.f., the parameter driving the error of the cohesive law is the flexural rigidity *EI*, but as the number of traction d.o.f. increases, the error shows a higher correlation with the shear rigidity *kAG*. For big amounts of traction d.o.f., the calculation is highly affected by the amount of *DIC noise*. This effect was attributed to the ill-conditioning of the system in the 3rd semester project Viejo et al. [2019]. A deeper study is conducted in Chapter 5.

For illustration purposes, and for avoiding displaying the correlation graphs for all the number of traction d.o.f used, another parameter is defined. This parameter is the area A_k under the PPMCC curve, which is normalized by the maximum separation value δ_N.

$$A_k = \frac{\int_0^{\delta_N} |r_j(R_{ik}, \sigma_{ij})| d\delta}{\delta_N} \tag{4.5}$$

The parameter A_k can be used analogous to a correlation parameter to assess a general correlation between an input parameter and the calculated cohesive law $[\sigma_{ij}, \delta_j]$. Although it does not provide information about which parts of the cohesive law are more influenced by what parameters, it shows the general tendency of the influence of the input parameters R_{ik} in a single figure . It varies from 0 to 1, meaning 0 no correlation and 1 total linear correlation. The integral in Equation (4.5) is performed making use of the trapezoidal integration rule, with the function *trapz* in MATLAB. The variation of A_k with the number of traction d.o.f. is displayed for each parameter in Figure 4.12.

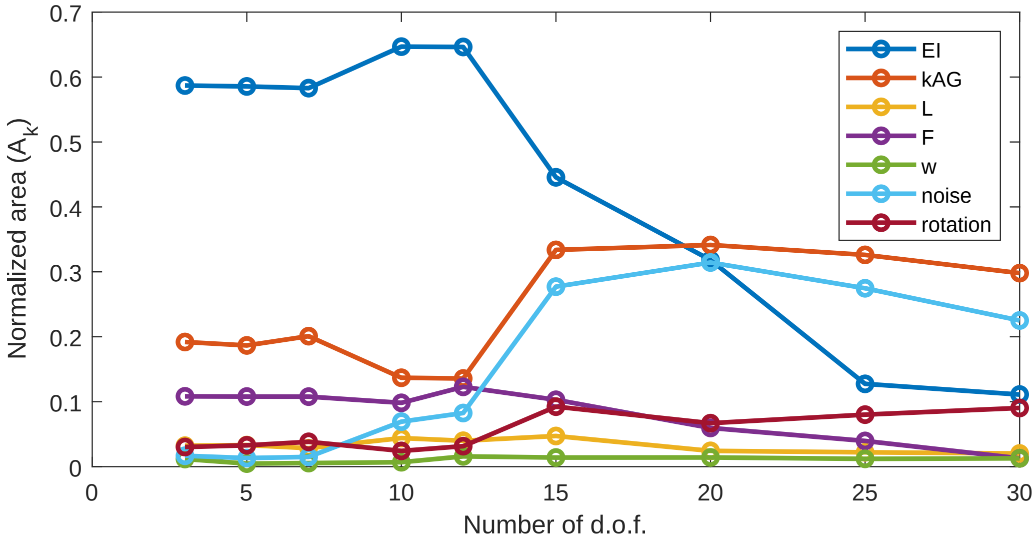


Figure 4.12: Normalized area A_k for each input parameter and number of traction d.o.f. when the initial guess of the nonlinear optimizer is the whole beam. The circles correspond to the calculated points.

For a low number of traction d.o.f., the most influencing parameter is the flexural rigidity *EI*. As the number of traction d.o.f. increases, the shear rigidity *kAG* becomes the dominant parameter, together with the DIC camera noise. The effect of the shear rigidity is assumed to happen because a higher number of triangular loads allow for more abrupt changes in the traction field along the length of the beam, which is directly related to the effect of the shear load on the displacements, and so, the shear rigidity. And the relation to the *DIC noise standard deviation* is related to the ill-conditioning of the system, this effect has been acknowledged in the previous semester project Viejo et al. [2019], and it is further studied in Chapter 5.

To check the results of the correlation analyses, two MC simulations are run: one reducing the uncertainty of *EI* and the other reducing the uncertainty of *kAG*. Thereby, it can be evaluated if they are the most influential over the results, by checking the correlation results. The uncertainties in Table 4.1 corresponding to the flexural rigidity *EI* and shear rigidity *kAG* are reduced by a number of 3, so $u(EI) = 1.15$ Nm^2 and $u(kAG) = 15400$ N. The results are displayed in Figures 4.13 and 4.14.

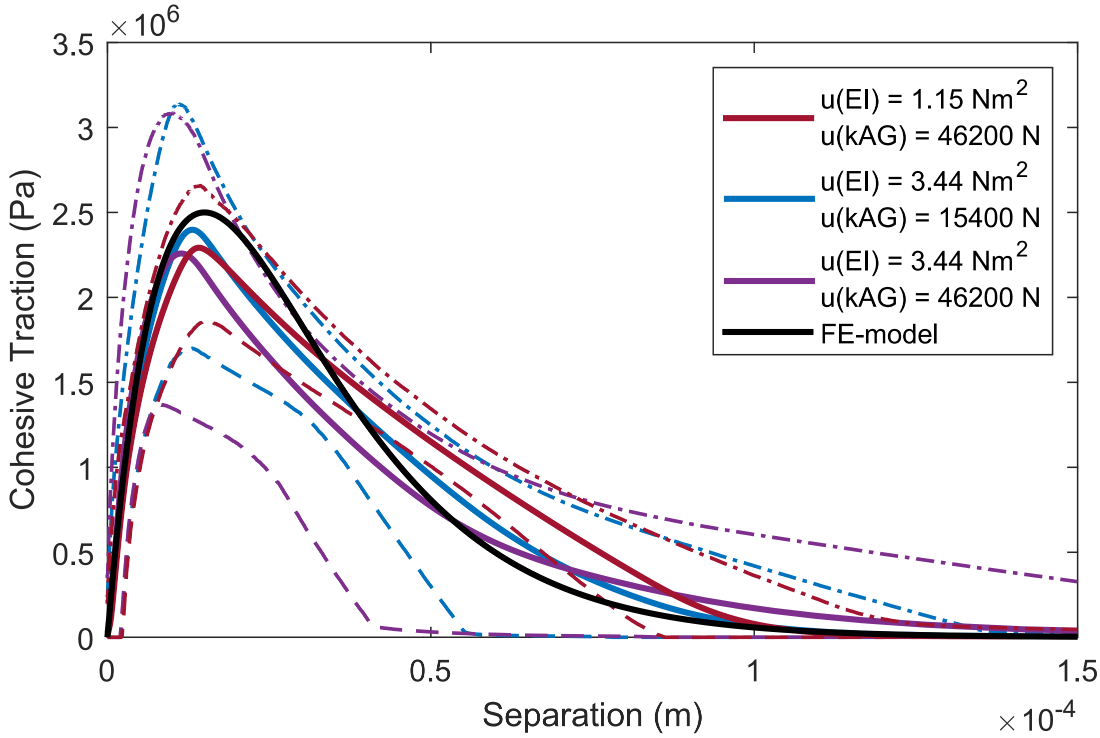


Figure 4.13: Resulting mean cohesive laws $[\overline{\sigma}_j^M, \delta_j]$ (continuous line) and confidence interval, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{UB,j}^M]$ (dashed-dot line), obtained from the MC algorithm using 10 traction d.o.f. The legend shows which curve correspond to which change in the uncertainty of the input.

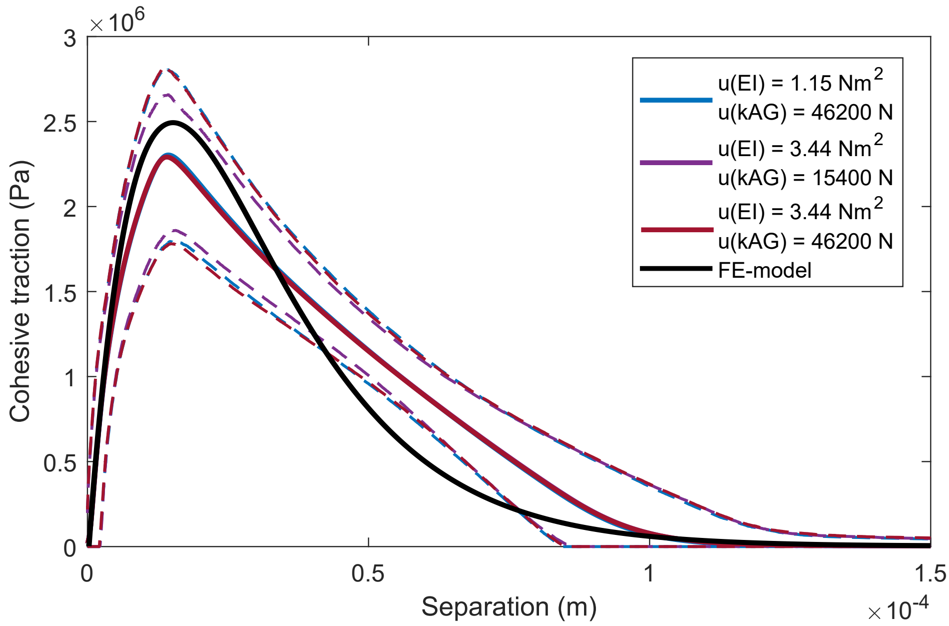


Figure 4.14: Resulting mean cohesive laws (continuous line) and confidence interval, delimited by the bounds (dashed lines), obtained from the MC algorithm using 25 traction d.o.f. The legend shows which curve correspond to which change in the uncertainty of the input..

In Figure 4.13, the confidence interval of the cohesive law with lower $u(EI)$ is clearly narrower than the confidence interval of the original cohesive law. The curves with lower $u(kAG)$ show little difference with original. This two results are in agreement with Figure 4.12, which for 10 traction d.o.f., shows that *EI* is the parameter with most influence.

In Figure 4.14, the curves with lower $u(EI)$ are closer to the original curves, which is again in agreement with Figure 4.12. Moreover, the separation values where the difference is more noticeable correspond to the area where the correlation coefficients are higher in Figure 4.11.

This results support the reliability of the correlation analyses as a tool to find the parameters driving the error in the cohesive law.

4.3 Conclusions on the Monte Carlo Algorithm Results

The MC algorithm explained in this chapter is used as a tool to assess the sensitivity of an inverse parameter identification routine against the uncertainty of the input parameters by providing a confidence interval for the calculated cohesive law. Therefore, any change in the experimental procedure or the inverse parameter identification algorithm can be evaluated easily to check whether it represents an improvement or not.

From the result in Figures 4.7, 4.8 and 4.9, it has been seen that increasing the number of traction d.o.f. makes the mean curve get closer to the FE-model curve and the confidence interval to get narrower. However, after a certain amount of traction d.o.f. the system becomes more unstable, and the precision of the calculation decreases resulting in a wider confidence interval. Thereby, it can be concluded that the method needs from a trade-off between precision and robustness regarding the number of traction d.o.f. There exists an optimum number of degrees of freedom in terms of sensitivity to the uncertainty of the input.

The correlation tools presented in Appendix C have been proven to be useful at identifying the parameters with most influence on the uncertainty of the results: the flexural rigidity *EI* for low amount of traction d.o.f., and the shear rigidity *kAG* and the *DIC noise standard deviation* when a higher number of traction d.o.f. is used. These results, together with the outcome of MC, allow to redesign the inverse parameter identification model, as shown in Chapter 6, and to design the experimental procedure, as shown in Chapter 7.

Prior making the changes to improve the performance of the methodology, it is decided to benchmark the mathematical tools used to solve the inverse problem, which allows to calculate the tractions at the interface. A benchmark of this mathematical tools is explained in Chapter 5. This ensures that the numerical routines work as intended and that the most is made out of the implemented algorithms. In addition, any change on the inverse parameter identification model can be evaluated efficiently by repeating this benchmark process, as it is done in Chapter 6.

5 | Benchmark of the Inverse Parameter Algorithm

In this chapter, the studies done on the model after evaluating the results of the initial Monte Carlo investigation are explained. A study of the linear least squares system is done with the tools explained in Appendix B, which help to characterize the unstable behaviour of the model. After this, the study is translated towards the nonlinear optimization subroutine of InTraFiCa and its impact on the inverse parameter identification solution sensitivity.

5.1 Study of the Ill-Conditioning and Sensitivity of the System

Monte Carlo results have suggested that the linear system of equations used in the least-squares solution might need a further study, to address some of the instabilities that have appeared with the fluctuation of some of the inputs. In the developed inverse parameter identification algorithm, there has always been two user-defined variables that have a critical effect on the results obtained: *the number of triangular traction d.o.f used* for building the solution and *the bounds* for the location of the traction d.o.f. used. Note that the number of traction d.o.f. corresponds to the number of columns of the coefficient matrix G. The main difficulty of this problem rests on the fact that each time one of these parameters change, the linear system of equations to be solved changes, which means that the solution desired might not always be the best numerical solution, depending on the characteristics of the system. Therefore, for each combination of the two variables defined, the effect of the variability of the quantified uncertainties in Chapter 3 on the final result can pose a big difference from one solution to another.

The objective of this study is to determine whether a certain number of traction d.o.f., bounded between two certain positions, can produce an accurate and stable solution independently of the variability of the uncertainties associated to the calculations. For this study the concepts and tools introduced in Appendix B are used.

5.1.1 Effect of the Number of Traction d.o.f. on the Solution (σ, δ)

As mentioned in Appendix B, the magnification of the noise on the input is system-dependent. The first step, is to identify the type of system to be studied: *rank deficient* or *discrete ill-posed problem* (defined in Section B.3.1 in Appendix B). For this reason, systems with different number of traction d.o.f. (from 5 to 20) are analysed using the Singular Value Decomposition (SVD).

The SVD of the different coefficient matrices (from 5 traction d.o.f. until 20 d.o.f.) gives the progression of singular values presented in Figure 5.1, for the different number of traction d.o.f. located along the entire length of the beam.

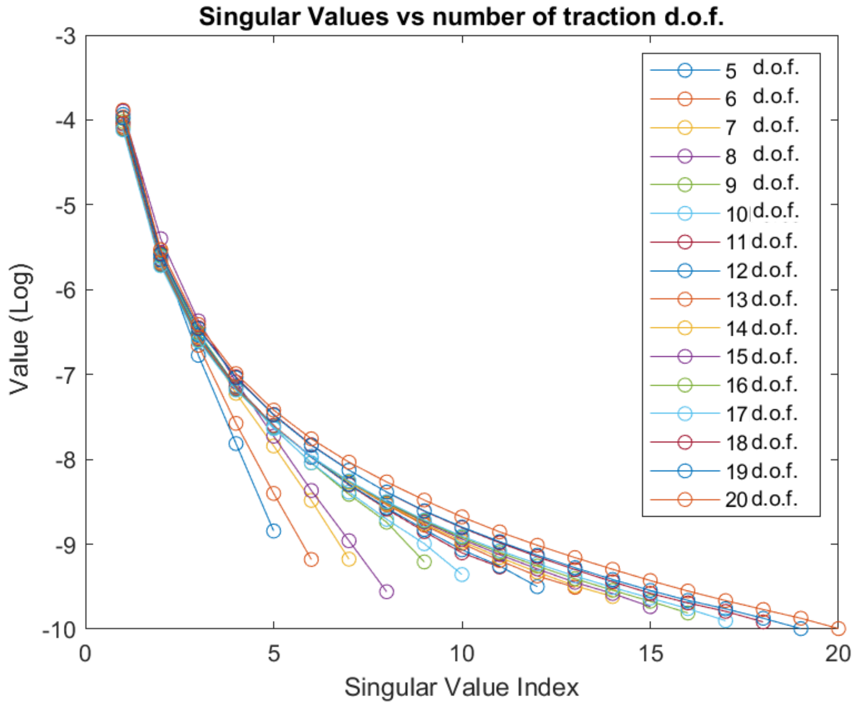


Figure 5.1: Singular values for the different number of degrees of freedom selected for the analysis. Note that the vertical axis follow a logarithmic scale.

It can be clearly seen how the singular values start approximately at the same level, independently of the number of traction d.o.f., and decay to zero. The decay process seems to be more gradual when the amount of singular values increase. This graph is an indicator that the problem that is being solved, is a *discrete ill-posed problem*. To ensure completely the previous statement, it is chosen to study the left and right singular vectors of the matrices.

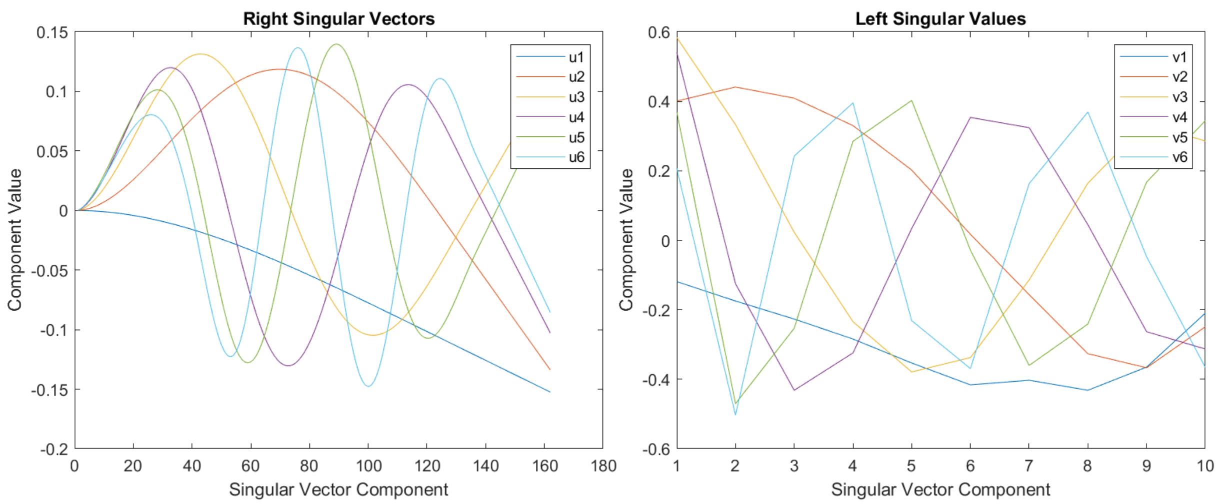


Figure 5.2: Left and right singular vectors extracted from the SVD for 10 traction d.o.f.

It is clearly seen in Figure 5.2, that the right and left singular vectors tend to have more sign changes as the index increases, which indicates that the problem is an *ill-posed problem*. Note that just the singular vectors for the coefficient matrix corresponding to 10 traction d.o.f. are plotted, for a clear display of information, but this trend is repeated for all the cases studied.

The next study concerns the degree of instability that each of these systems have when it comes to the error magnification of the noise in the input vector (vector d following the notation introduced in Appendix B). The parameter used to measure instability is the condition number of matrix G, which is related to the error bounds of the solution (explained in Section B.3.4 in Appendix B). The evolution of the condition number of each matrix G with respect to the number of traction d.o.f. used for the computation of the solution is shown in Figure 5.3.

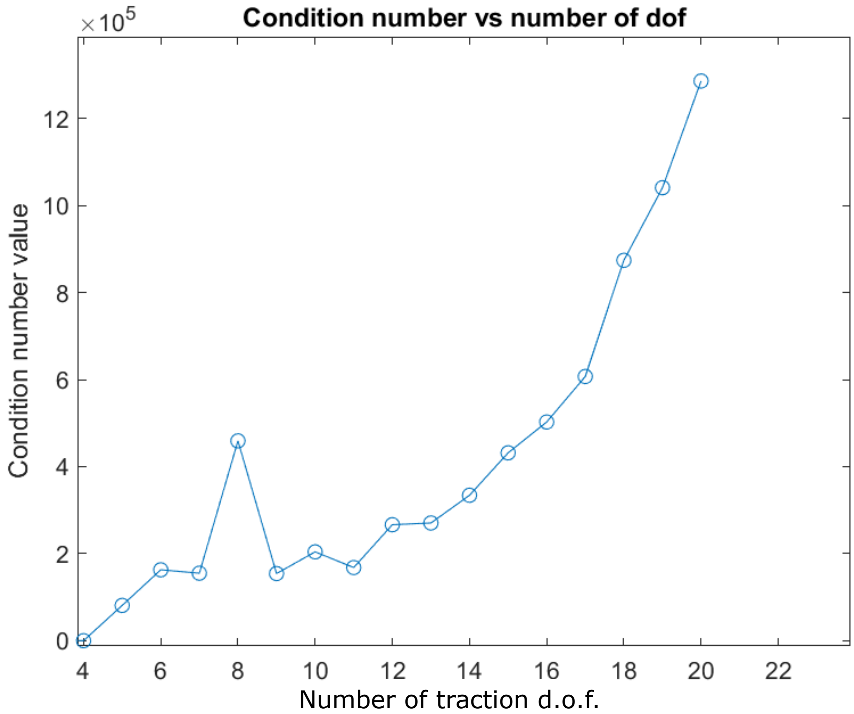


Figure 5.3: Condition number variation with the increase of the number of traction d.o.f. of the solution.

Notice that the condition number value maintains a certain stability from 5 to 13 traction d.o.f. approximately (excluding the case of 8 traction d.o.f.), and then it increases when more traction d.o.f. are used for solving the system. If the solution of each system is showed for the same input data, the obtained results (see Figure 5.4) are relatively close to each other (excluding the ones for 18, 19 and 20 traction d.o.f.). This indicates, that when a good solution is achieved, adding more traction d.o.f. does not improve the outcome of the result, but just makes the system more sensitive to the noise introduced, obtaining a different solution.

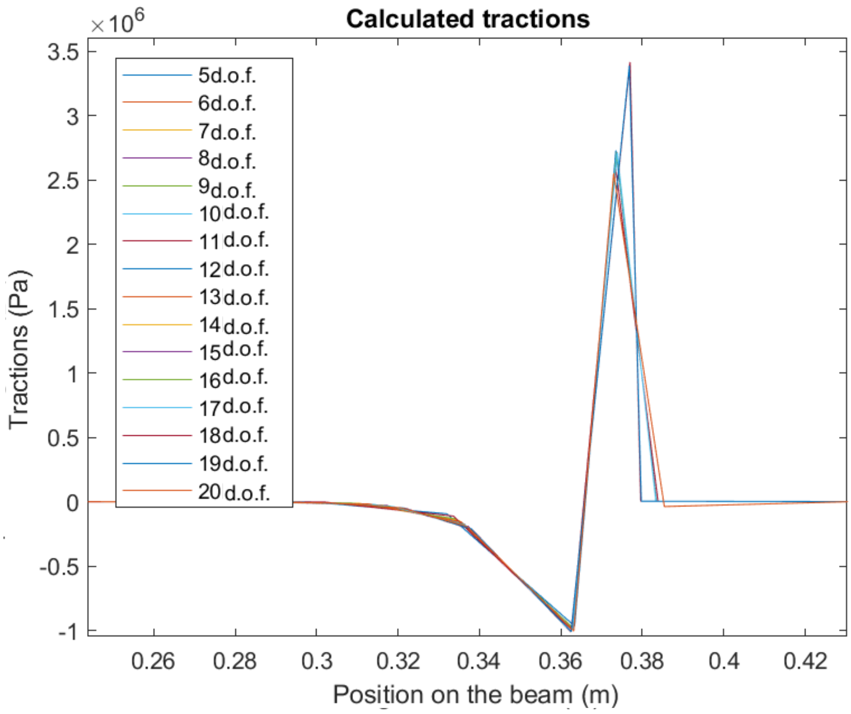


Figure 5.4: Calculated tractions for different number of traction d.o.f.

Moreover, a guess of how many degrees of freedom are optimal to achieve the best solution possible with the load functions that have been proposed can be made evaluating the norm of the residual of the final system. The norm of the residual is calculated (using the formula shown in Equation (B.3) in Appendix B) for the solution of the system for a various amount of traction d.o.f. (from 3 to 15).

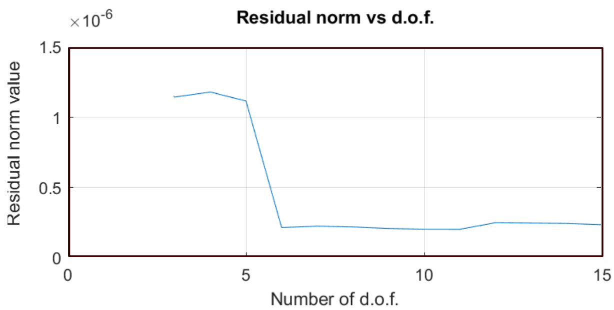


Figure 5.5: Value of the norm of the residual of the solution proposed vs number of traction d.o.f. used in the algorithm.

In this case, as shown in Figure 5.5, it can be seen how the minimum value of the norm is reached after using 6 traction d.o.f. After this number, the solution is not more accurate, as shown in Figure 5.4, so adding more traction d.o.f. is not necessary. However, for very few traction d.o.f. the system cannot reach a realistic solution due to the limitations on the shape of the traction functions that are used.

5.1.2 Effect of the Bounds for the d.o.f. on the Solution (σ, δ)

With the effect of the number of traction d.o.f. presented, the stability of the system with the variation of the space allocated to the load positioning is analysed. Note that for a successful calculation the limits established for the distributed load obviously have to include the fracture process zone.

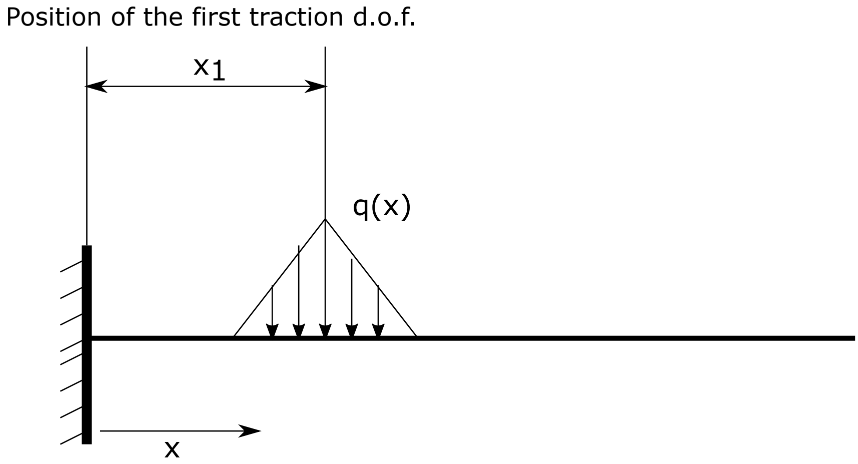


Figure 5.6: Position x_1 is the variable studied.

Therefore, the study is performed with the computation of the singular values of the system's coefficient matrix G for different locations of the first triangle peak (represented in Figure 5.6). The study is done with 8 possible positions for the initial load (0.05m, 0.0786m, 0.1071m, 0.1357m, 0.1643m, 0.1929m, 0.2214m, 0.25m) as from 0.25m on the FPZ is outside of the bounds and tractions are not computed correctly. Note that only the position of the first traction d.o.f varies, as it has been observed how the impact of the bound of the last traction d.o.f. does not impact the solution obtained. The study is made with 13 traction d.o.f, proven to be the system that offered a good stability, as shown in Figure 5.3, with a number of traction d.o.f. that allows for a good representation of the interfacial tractions.

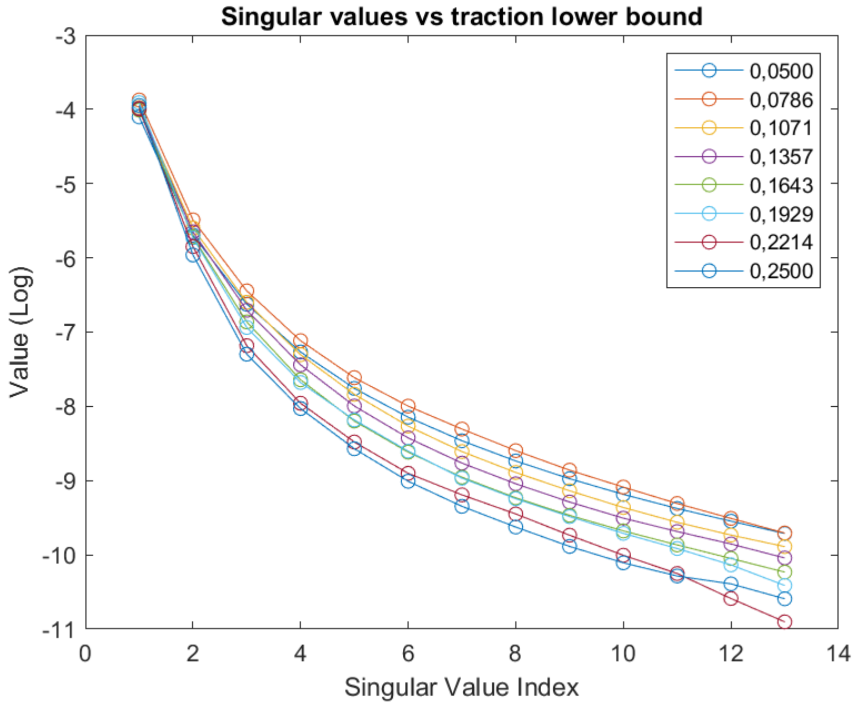


Figure 5.7: Singular values progression for different positions of the lower bound. The legend shows the distance of the first load to the clamp in meters.

Observing the evolution of the system singular values in Figure 5.7, it is clear from the theory introduced in Section B.3.1 in Appendix B that the problem remains a *discrete ill-posed problem*. Regarding the condition number for the different cases studied, Figure 5.8 shows that the condition number tends to increase when the loads are more bounded. This behaviour is expected, as the system is forced to use more d.o.f. than it would for describing the traction shape, making the system more unstable. This effect can be seen as the singular values in Figure 5.7 decrease in value as the traction d.o.f. bound is smaller, making the solution more unstable than when the traction d.o.f. are not bounded.

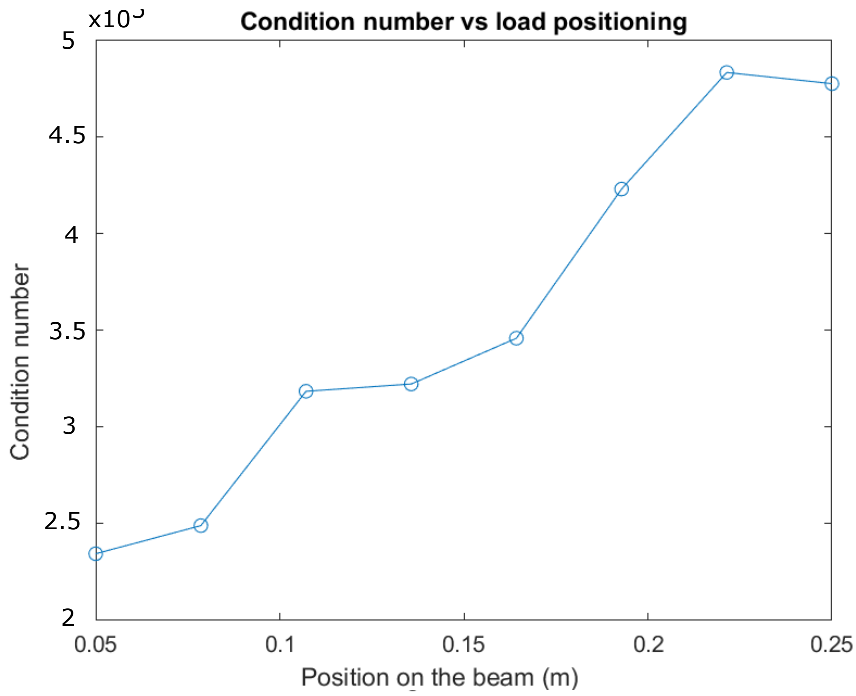


Figure 5.8: Condition number evolution with different locations of the first traction d.o.f.

It is seen in Figure 5.9, how the level of accuracy of the solution seems to be higher (as it is known the maximum tensile stress should be around 2.5 MPa) when the fracture process zone is well bounded or the whole length of the beam is available for the system to place the loads. Even though it makes more sense to believe that a well bounded location for the loads would make the system much more stable and more precise than with the full length of the beam, the results show otherwise.

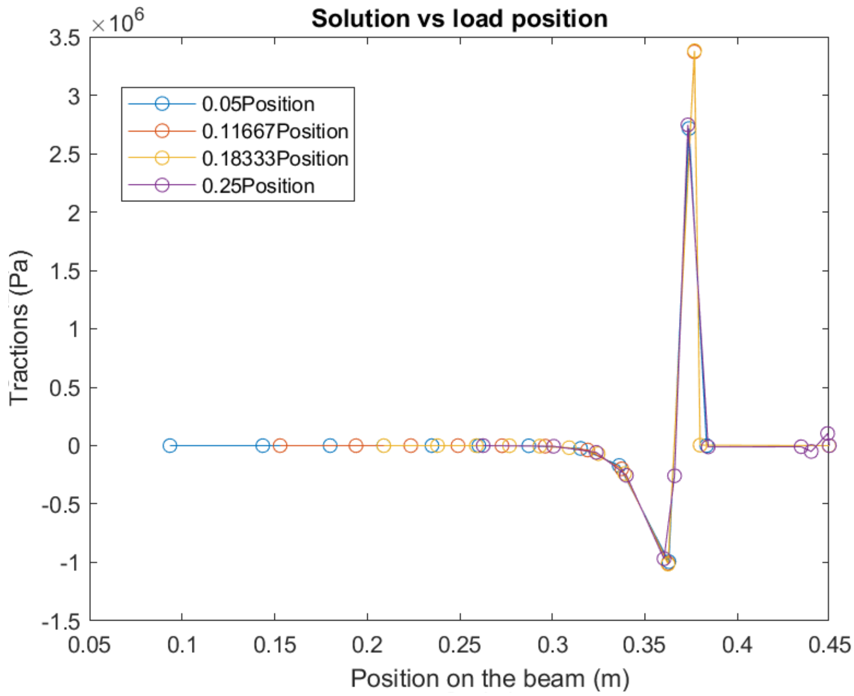


Figure 5.9: Solutions plotted for different pre-defined bounds of the first traction d.o.f.

5.1.3 Noise Effects on the Results (σ, δ)

Theory introduced in Section B.3.4 in Appendix B shows that the ill-conditioning of the problem creates an issue of noise magnification when the displacement used as input for the system has a relatively small perturbation. For the best combination of number of traction d.o.f. and an optimum selection of the traction d.o.f. bounds for their position on the beam, the condition number values are of the order of magnitude of 10^5. The proof of the noise magnification due to the ill-conditioning of the coefficient matrix can be seen for 13 traction d.o.f. and a random normally-distributed noise (with an order of magnitude of 10^{-6}) introduced in the input displacement field obtained from FE simulation in Figure 5.10. Note that the real world noise values obtained from the noise floor analysis of the DIC is of the order of 10^{-7} as shown in Section 3.2.

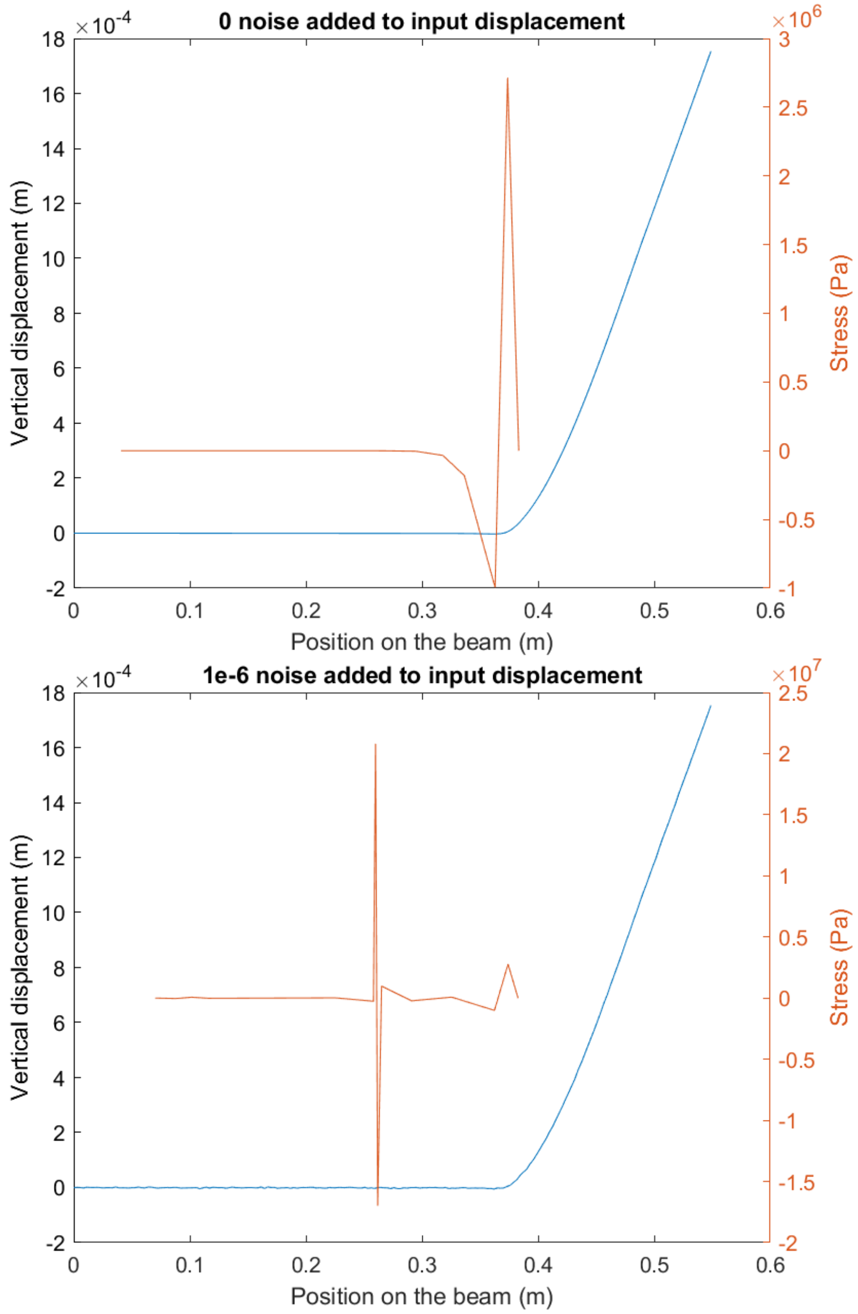


Figure 5.10: Effect of different noise components introduced on the input displacement on the computed solution of the system.

In this case, following the theory in Section B.3.5 in Appendix B, the stabilization of the system can be achieved with the addition of regularization to the problem. In this case the generalized Tikhonov regularization is applied to the minimization problem corresponding to the noise of magnitude 1e-6 (second case in Figure 5.10).

The first step before the computation of the solution is to find the optimum λ factor via the L-curve approach (explained in Section B.3.5 in Appendix B). It is clearly seen in Figure 5.11, that the optimum value of λ for the regularized solution is given by area around the corner of the L-curve (highlighted in red). Note that the final value of lambda is exclusively dependent on the judgement of the user to achieve the desired output.

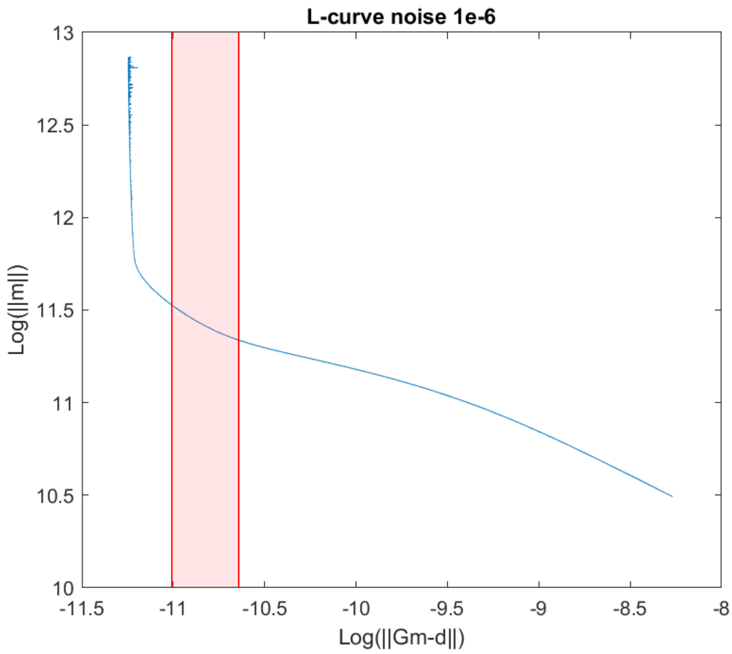


Figure 5.11: L-curve for the present problem with the possible area of lambda values to choose shadowed in red.

After solving the system with regularization, using $\lambda = 3.3646 \times 10^{-10}$, the solution obtained is displayed in Figure 5.12. It is demonstrated that regularization is a powerful and valid tool that can help the stabilization of the solution, but without the real solution being embedded on the final system's response it is not of any use. Moreover, this method is completely dependent on user's choice and has no physical meaning, so it is not used further in this project.

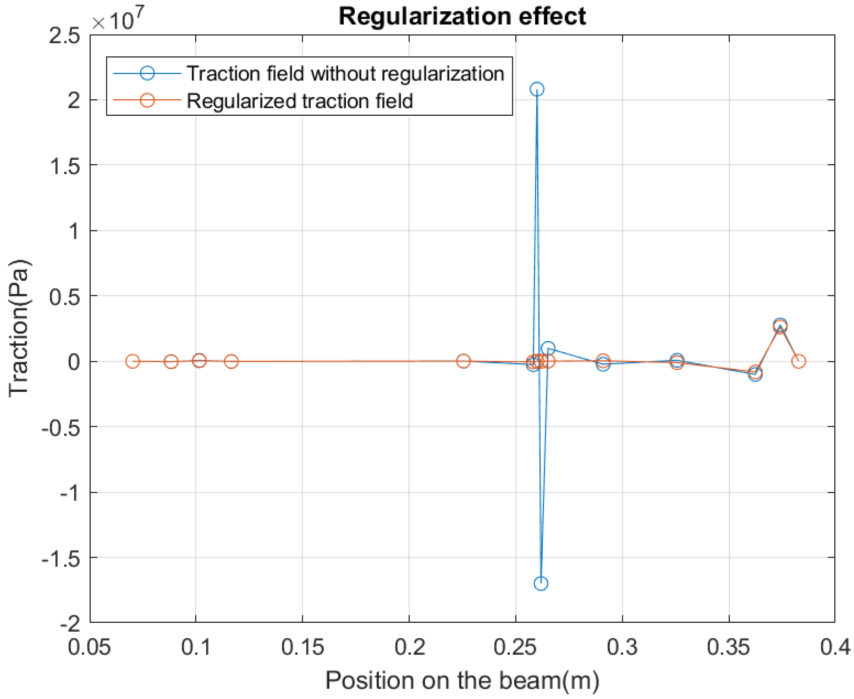


Figure 5.12: Regularization effects on the system's response.

5.1.4 Effect of Nonlinear Optimization on the Coefficient Matrix G

So far, the system of equations that the nonlinear optimization gives as output has been analysed. However, the linear least-squares solution is embedded in the nonlinear optimization problem used for locating the traction d.o.f. at a certain position on the beam. The function f in Equation D.14 in Appendix D, used to build the coefficient matrix G of the system, depends on the position $\mathbf{x}$ of the d.o.f., which is the output of the nonlinear optimizer. This means, that with each function evaluation of the nonlinear optimizer, the systems coefficient matrix changes, solving a different linear least-squares problem that might have a different solution (this topic is treated in detail in Section 5.2). However, even though the output of the nonlinear optimizer is the optimal positioning $\mathbf{x}$ of the traction d.o.f., the coefficient matrix might be so sensitive to the noise that the final system can be of no use for the purpose of this project.

Therefore, it is chosen to evaluate the condition number variation with each function evaluation of the nonlinear optimizer to see whether, and to what extent, this process affects the characteristics of the linear system (Figure 5.13).

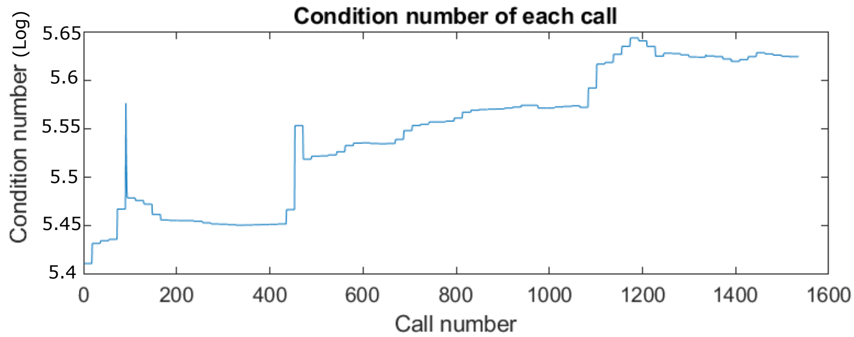


Figure 5.13: Evolution of the condition number of the system with the number of function evaluations.

The results are shown in Figure 5.13 just with 13 traction d.o.f. for simplicity, although the same trend is observed in the rest of the cases. It is clearly seen in Figure 5.13, how the condition number increases with the number of function evaluations. That is to say, the optimum load positioning makes the solution accuracy much better than in the case where all the loads are separated the same (initial guess of **x** for the location of the traction d.o.f.), but gives a more unstable system.

5.1.5 Conclusions on the Linear Least-Squares System Behaviour

After the evaluation of the linear system's behaviour the general picture is clear. First of all, it seems that the instabilities of the system arise from the formulation of the coefficient matrix. The beam model created has too many parameters that strongly influence the coefficients of the system's matrix.

The main conclusion is related to the capabilities of the system. It seems from the results that the limitations of this formulation have been reached. Adding more traction d.o.f. to the system, does not make the solution to be better or to be more stable, but completely the opposite. Moreover, different displacement input ranges, or traction bounds are not causing the solution to change or the system to be robust with respect to input noise.

In addition, it has been demonstrated how regularization can be useful up to a certain extent and used if the noise of the input displacement is relatively significant. However, it is a tool that needs the user to know what is the response that is wanted to be obtained, which is not the case in this project, where the cohesive law of the material is unknown. It is not a tool to use to enhance the own characteristics of the system and it is strongly dependent on the regularization parameter which does not have any physical implication. Therefore, even though it is a valid instrument to use under certain situations, it is believed that the linear system can be improved further to give a better solution.

Finally, the nonlinear optimization has been found to be a part of the problem, due to the fact that the linear system changes with every iteration, so a "different" problem is obtained with each call. A minimum number of traction d.o.f. is needed for the nonlinear optimizer to find the coefficient matrix that gives the minimum norm of the residual possible. However, it has to be noted that the increase in the condition number due to the nonlinear optimization process is never as big as the increase due to the first two variables studied in the previous sections, i.e. number of traction d.o.f. and bounds for the location of the traction distribution.

The findings obtained suggest that a deeper understanding of the nonlinear optimization effects on the system must be achieved, which is the motivation for the following sections.

5.2 Analysis of the Nonlinear Optimization Subroutine of *InTraFiCa*

In a nonlinear optimization problem, either the objective function, the constraints, or both, are nonlinear. The nonlinear optimization subroutine, presented in the previous semester project [Viejo et al., 2019], consists in the calculation of the distributed tractions which generates the same displacement field as the one given as input to the inverse parameter identification tool. Equations (5.1) and (5.2) present the objective function and the matrices (A and B) of the inequality constraints respectively. The design variables $\mathbf{x}$, are the position of the distributed traction d.o.f. In the present problem, the cost function is the Euclidean norm of the residual ($\|r\|$) of the displacements (d). The relation between the design variables $\mathbf{x}$, and the coefficient matrix G, is nonlinear (see Equation (D.6) in Appendix D). On the other hand, the constraints present are inequality constraints of linear nature. Thereby, the present optimization problem is nonlinear and constrained.

$$
\begin{aligned}
&\underset{(\mathbf{x})}{\text{Minimize:}} && \|r\| = \|[G(\mathbf{x})]\left([G(\mathbf{x})]^T[G(\mathbf{x})]\right)^{-1}[G(\mathbf{x})]^T\{d\} - \{d\}\|_2^2 \\
&\text{Subject to:} && A\mathbf{x} \leq B
\end{aligned}
\tag{5.1}
$$

$$
A = \begin{pmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\quad , \quad
B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ L \end{pmatrix}
\tag{5.2}
$$

As it has been pointed out in Chapter 4, when the positions of the peaks of the traction d.o.f. i.e. the design variables $\mathbf{x}$, are placed in the zone where the interfacial tractions are nonzero, the obtained cohesive law approximates much better the one that is aimed. However, the exact location of the interfacial tractions is practically unknown in a real experiment. The nonlinear optimization problem solves this issue, as it redistributes the initially placed design variables along the length of the beam. As it has been shown in Figure 5.13, the final configuration of the coefficient matrix G, depends on the outcome of the nonlinear optimization subroutine. Besides, the previous section presented the influence of the nonlinear optimizer on the condition number values. This is reflected in the stability of the linear least-squares system and the performance of the inverse parameter identification tool.

This sets the ground for the nonlinear optimization study, which is presented in the following paragraphs. The aim of this work is to evaluate the performance of the different algorithms that are available and to see how the results are related with the distribution of the loads and the condition number values. First, an overview of nonlinear optimization and the tools available is given. Then, two studies are performed. One to assess the values of the tolerances for an optimum performance of the algorithm; the second, takes into account the effect of the region where the design variables can be allocated.

The current problem is solved using MATLAB by means of *fmincon*, which is a built-in function capable of dealing with constrained nonlinear optimization. The algorithms available in the *fmincon* subroutine are stated in the following.

- Interior-point.
- Sequential quadratic programming.
- Trust-region-reflective.

These algorithms pertain to the class of the gradient-based methods, which use first and/or second order derivatives of the objective function, to check if a given point is a local minimum [Arora, 2017]. Therefore, it is assumed that the objective function at hand is continuous and twice differentiable.

The trust-region-reflective algorithm is dismissed, since its formulation does not admit inequality constraints. Therefore, the options which are left are the *interior-point* or the *sequential quadratic programming (SQP)* algorithms. Both of these algorithms follow the line search strategy, where a descent direction and a step size, which minimize the current function value, are calculated at each iteration [Nocedal and Wright, 2006]. A brief explanation of the *interior-point* and the *SQP* methods is given in the following paragraphs. The information is extracted from Arora [2017] and Nocedal and Wright [2006], wherein a thorough explanation of the algorithms can be found.

Sequential quadratic programming

The *SQP* method is a two-step algorithm. The first step consists in the calculation of the descent direction, also known as the search direction subproblem. This subproblem is of quadratic nature since a second order term is added to the formulation of the objective function. The second order information is given by the Hessian, which is approximated with a Quasi-Newton method. Then, the objective function is reformulated via the Augmented Lagrangian Method, and an unconstrained formulation is attained. Finally, the search direction is found by solving the Karush-Kuhn-Tucker (KKT) conditions at each iteration [Lund, 2019]. The second step or subproblem is the step size calculation. It is performed by minimizing the augmented cost function in the descent direction, obtained from the first subproblem. A line search method e.g. golden section or polynomial interpolation, is used.

Interior-point

The *interior-point* algorithm is a line search method which is characterized by solving a sequence of equality constrained subproblems, called barrier subproblems, instead of the original inequality constrained problem [Waltz et al., 2006]. The objective function is expanded with an extra term. This term is the barrier function, and its objective is to penalize and restrict the augmented cost function when a constraint is approached.

The unconstrained subproblems are obtained from the application of the Augmented Lagrange Method. Thereafter, by writing the first-order optimality conditions, a system of equations is posed. It is known as the *primal-dual* system, which is solved in each iteration using linear algebra. If during the solution process of the primal-dual system, the algorithm determines that the Hessian is not positive definite, the algorithm changes the approach to a trust-region method which uses a conjugate gradient step [Waltz et al., 2006].

5.2.1 Algorithm Selection

Since two options are available to solve the nonlinear problem, it is desired to compare the performance of both algorithms presented above. In order to do that, the values of the objective function when a candidate point is found are observed and compared. Different problems are solved by changing the values of the number of traction d.o.f., but also for distinct initial guesses of the design variables distribution. Therefore, it is desired to check whether the algorithms converge to the same optimum or not. The preferable case, is the one where the same optimum is obtained for the different initial points, which might indicate the convexity of the problem [Lund, 2019].

Tables 5.1 and 5.2 display the results from the first study, where the algorithms are applied with the default options. The analyses have been performed for 5, 7, 10, 15 and 20 traction d.o.f. Table 5.1 presents the first results obtained using the *interior-point* algorithm. The values of the objective function at the candidate points obtained from different initial guesses for the design variables $\mathbf{x}$, are shown for each number of traction d.o.f., i.e. each distinct problem. The initial guess sets the zone where the design variables $\mathbf{x}$ are firstly placed and equally spaced.

Algorithm	Interior-Point				
Number of traction d.o.f.	5	7	10	15	20
Initial guess area (mm)	Objective function value ($\times 10^{-5}$)				
[0 , 0.485]	0.1535	0.1504	0.0148	0.0152	0.0122
[0 , 0.45]	0.1536	0.1504	0.0149	0.0152	0.0122
[0 , 0.3]	0.1505	0.1481	0.0148	0.0152	0.0122
[0.1 , 0.485]	0.1534	0.0172	0.0173	0.0153	0.0123
[0.1 , 0.45]	0.1545	0.0171	0.015	0.0152	0.0118
[0.1 , 0.3]	0.1502	0.1479	0.0145	0.0151	0.0122
[0.2 , 0.485]	0.0263	0.1483	0.0148	0.0153	0.0122
[0.2 , 0.45]	0.1533	0.0204	0.0149	0.0153	0.0105
[0.2 , 0.3]	0.1501	0.1506	0.0148	0.0153	0.0123

Table 5.1: Objective function values for different number of traction d.o.f. and initial guesses using the *interior-point* algorithm.

For a lower amount of traction d.o.f., the values of the cost function vary more among themselves. As the number of traction d.o.f. increases, the values of the objective function become more similar towards one value. One way to check if the optimization has succeeded is to plot the obtained load distribution for each initial guess. Figure 5.14 depicts the obtained load distributions for the case of 7 traction d.o.f., wherein the legend states the initial guess of each distribution. It can be seen that the distributions tend to capture the same shape; the traction d.o.f. have similar magnitudes and the design variables **x** are allocated mainly on the same position. Despite the initial guesses being substantially different, the results tend to the same distribution. However, two different peaks are detected in the negative part of the graph. These three distributions are the ones which have the lowest objective function values, as it can be seen in Table 5.1.

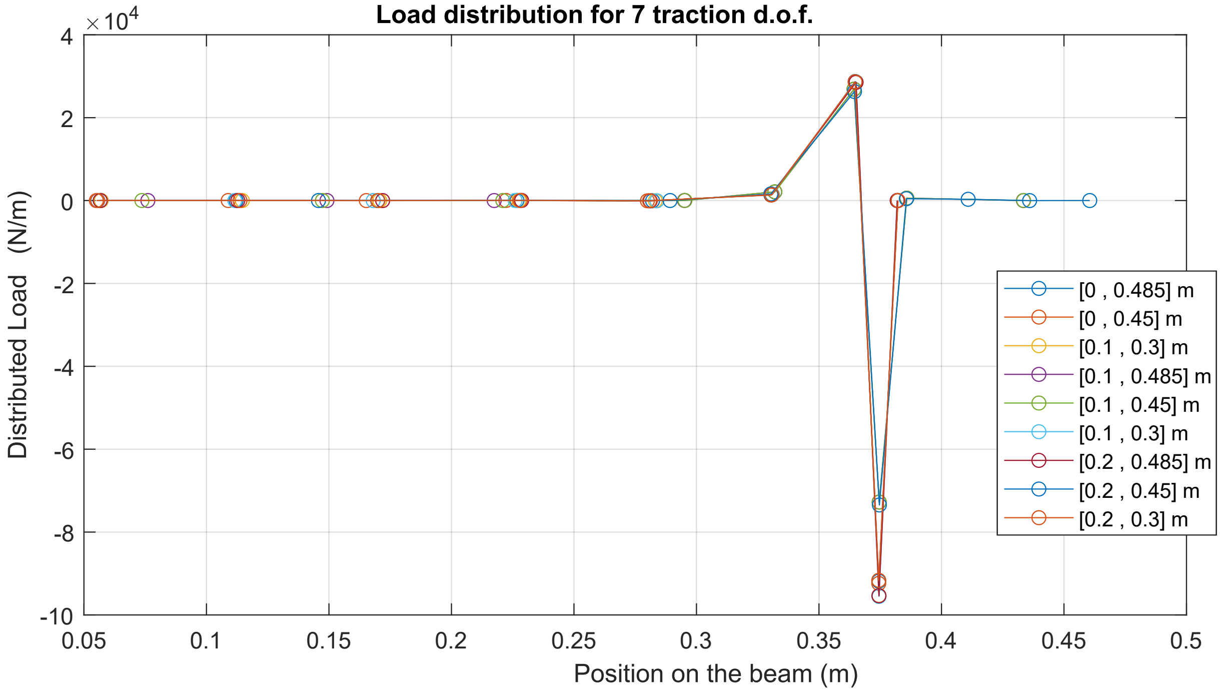


Figure 5.14: Load distributions obtained from nine different initial guesses, using 7 traction d.o.f.

On the other hand, the results obtained from the *SQP* algorithm with the default options are stated in Table 5.2. The same arrangement of the results as in Table 5.1 is followed.

Algorithm	SQP				
Number of traction d.o.f.	5	7	10	15	20
Initial guess area (mm)	Objective function value ($\times 10^{-5}$)				
[0 , 0.485]	0.1394	0.1246	0.0136	0.0112	0.0106
[0 , 0.45]	0.1739	0.1699	0.0139	0.0116	0.0122
[0 , 0.3]	0.1863	0.1661	0.1702	0.1422	0.1665
[0.1 , 0.485]	0.0189	0.0145	0.012	0.0109	0.0104
[0.1 , 0.45]	0.1663	0.0146	0.0126	0.0133	0.0104
[0.1 , 0.3]	0.1794	0.1694	0.1419	0.1401	0.1449
[0.2 , 0.485]	0.0164	0.0125	0.011	0.0104	0.0097
[0.2 , 0.45]	0.0162	0.0126	0.0121	0.0107	0.0098
[0.2 , 0.3]	0.1423	0.1663	0.1393	0.139	0.1389

Table 5.2: Objective function values for different d.o.f. and initial guesses using the *SQP* algorithm.

As opposed to the results from the *interior-point* algorithm, for each problem (different number of traction d.o.f.), the values of the objective function for the *SQP* algorithm differ much from one to each other. The load distributions for the problem with 7 traction d.o.f. are plotted in Figure 5.15. Clearly, there is a substantial difference between this figure and Figure 5.14. The prediction of both the traction d.o.f. and design variables **x**, is highly dependant on the initial guess, and the values that are obtained differ from the expected distribution e.g. a spike of $-8 \times 10^8\ N/m$ is detected. The graph on the right hand side of Figure 5.15 shows that some of the distributions are correct, but the performance is far lower than the one offered by the *interior-point*.

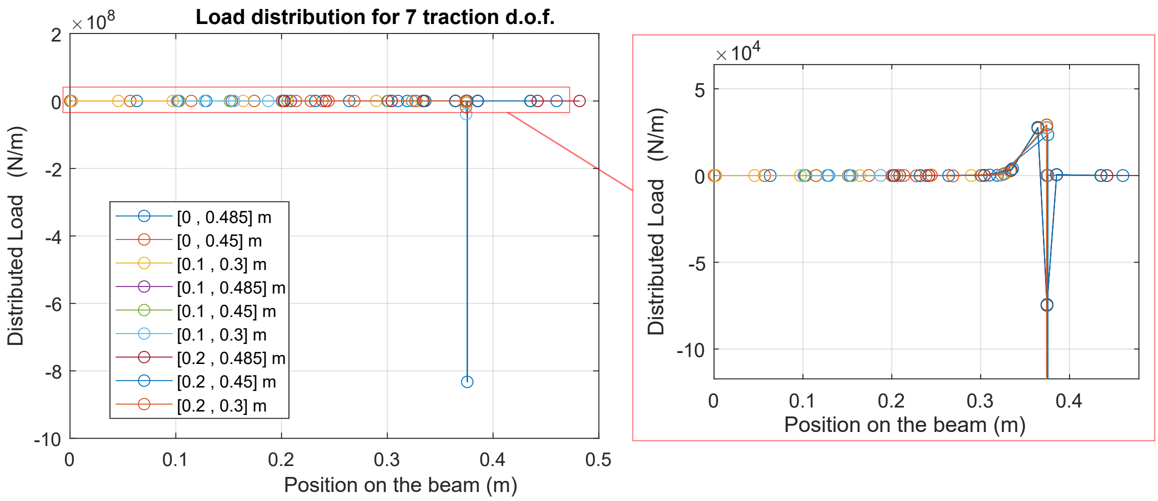


Figure 5.15: Load distributions obtained for 7 d.o.f., using *SQP* algorithm and nine different initial guesses.

As it has been stated previously in the present chapter, the value of the condition number plays an important role in the noise magnification of the linear least-squares system (see Appendix B). Due to the behaviour observed in the results from the *SQP* algorithm, it is believed that the result is a consequence of the instability of the system. Therefore, it is decided to evaluate the values of the condition number during the iterative process for both algorithms, see Figure 5.16. In this figure, the initial guess is $[0\,,\,0.485]$ meters.

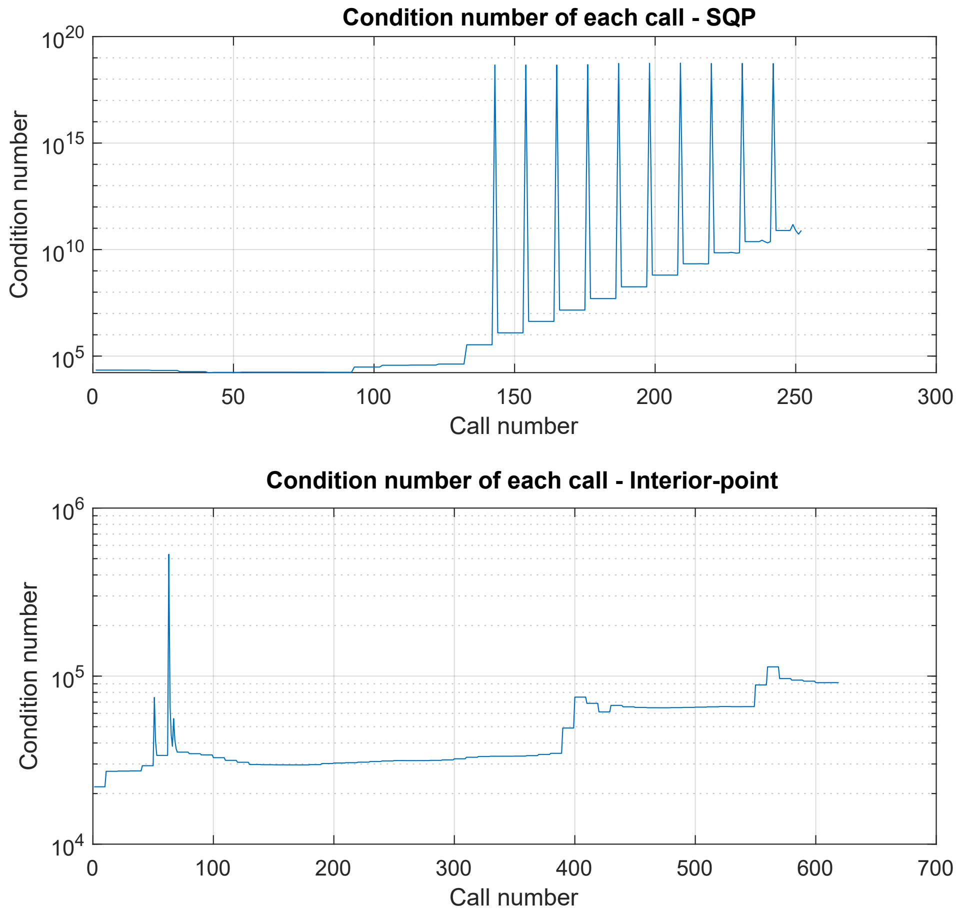


Figure 5.16: Values of the condition number at each function evaluation for the *SQP* and *interior-point* algorithms.

The top graph in Figure 5.16 displays the evaluation of the condition number of the coefficient matrix during the iterative process for the *SQP* algorithm. As suspected, the instabilities in Figure 5.15 are brought from the abrupt increase of the condition number values. The values skyrocket up to 10^{19}. As the iteration process finishes, the condition number of the coefficient matrix is of the order of 10^{11}. The same trend in the evolution of the condition number for the *SQP* is observed for different problems with different number of traction d.o.f.

Regardless of the initial guess, the *interior-point* shows a certain degree of robustness, as for the majority of the analyses, there is a common trend for the design variables to allocate in the same positions. Moreover, a plausible distribution is always achieved. However, the *SQP* displays a much different behaviour, wherein the load distributions have many different shapes, sometimes resulting in unfeasible distributions, as in Figure 5.15. It is seen that the algorithm is sensitive to the initial guess, since the results differ the most from the correct load distribution when the initial guess does not comprise the fracture process zone. Therefore, it is believed that the *SQP* algorithm must be abandoned in favor of the interior-point. The reason of the *interior-point* performing better resides in its formulation, as the barrier function prevents the objective function from approaching the imposed constraints i.e. the position of the traction peaks to get too close. An effect which has been seen to have a negative impact on the condition number value, shown in Figure 5.8.

5.2.2 Study of Tolerances and Bounds

As it can be seen in Table 5.1, the objective function values for an elevated number of traction d.o.f. are highly similar, while for a lower amount some differences are observed in the values. Note that these results are obtained with the default parameters of the optimization tool. Among these parameters, the stopping criteria are of particular interest. These criteria are based on two tolerances:

- Step tolerance
- Optimality tolerance

The first one constricts the smallest step size the algorithm can take in an iteration. The second one, takes into account the first-order optimality measure, which is meant to have zero value at the minimum. When one of the tolerances is met, the optimization process stops. For the case of 7 traction d.o.f., Table 5.1 suggests that more than one local minimum might have been found, since the objective function values are close to 1.5×10^{-6} or 1.2×10^{-7}. However, it cannot be assured that these values correspond to the global minimum.

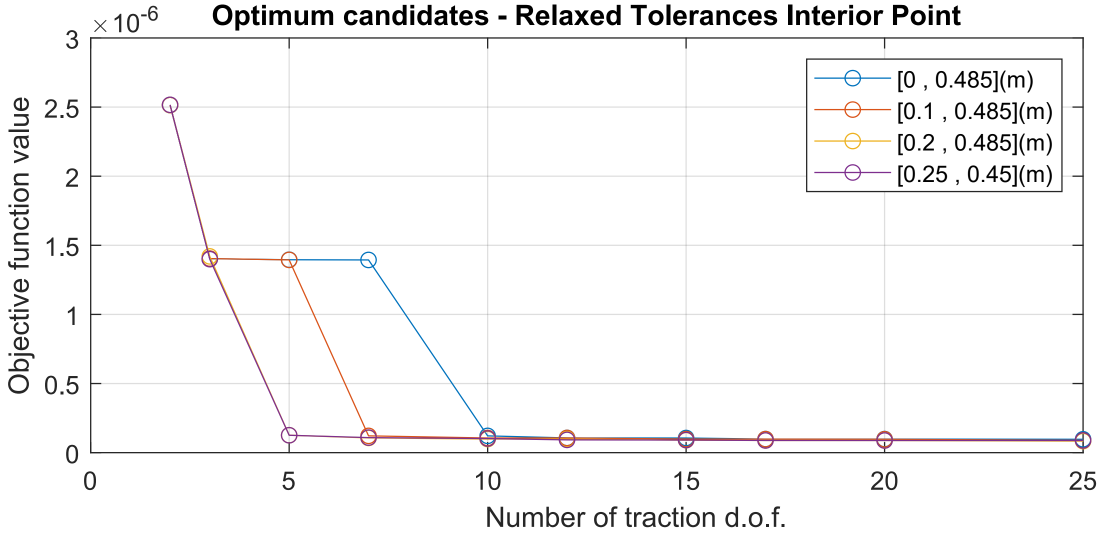
It is desired to know if more feasible points exist, for which the cost function values are lower than the ones shown in Table 5.1. Therefore, it is believed that reducing the values of the tolerances can result in the algorithm reaching lower objective function values. The procedure followed to reduce the tolerances is based on the output *exitflag*, given by *fmincon*. It states the reason why the iterative process stopped. Then, the tolerances are adjusted according to the values of *exitflag* until no difference between objective function values is observed with respect to the previous test. The following table shows the results for the problem of 7 traction d.o.f., when the tolerances are adjusted.

Number of traction d.o.f.	7			
Tolerances	Default	First reduction	Second reduction	Third reduction
Initial guess area (mm)	Objective function value ($\times 10^{-5}$)			
[0 , 0.485]	0.1504	0.1394	0.1394	0.1394
[0 , 0.45]	0.1504	0.1394	0.1394	0.1394
[0 , 0.3]	0.1481	0.148	0.148	0.148
[0.1 , 0.485]	0.0172	0.0123	0.0123	0.0123
[0.1 , 0.45]	0.0171	0.0123	0.0121	0.0121
[0.1 , 0.3]	0.1479	0.1393	0.1393	0.1393
[0.2 , 0.485]	0.1483	0.1393	0.1393	0.1393
[0.2 , 0.45]	0.0204	0.0105	0.0104	0.0104
[0.2 , 0.3]	0.1506	0.1394	0.1394	0.1394

Table 5.3: Objective function values with the tolerance reduction procedure.

The table shows that the reduction of the tolerance values is accompanied by lower values of the objective function. Despite these tests being only for one number of d.o.f., the same trend is observed in all the other analyses performed with different number of traction d.o.f. However, the difference between the values in Table 5.1 and the ones in Table 5.3 is remarkable. Stated before, it is observed that depending on the initial guess, two different optimum points are reached. This may indicate that two local minima are found. Figure 5.14 shows the result of reaching one minimum or the other, as the distributions of the traction peaks are distinct.

Stated before in Chapter 4, when the region where the design variables can be allocated encloses only the nonzero tractions zone, results are more precise. Thus, the response of the system when the design variables are bounded is also examined. The following image displays the objective function values for four different regions where the design variables are bounded. As it has been shown that the reduction of the tolerances has a positive effect on finding a lower value for the objective function, the analyses for the bounded regions are performed with the reduced tolerances. The results can be seen in Figure 5.17.



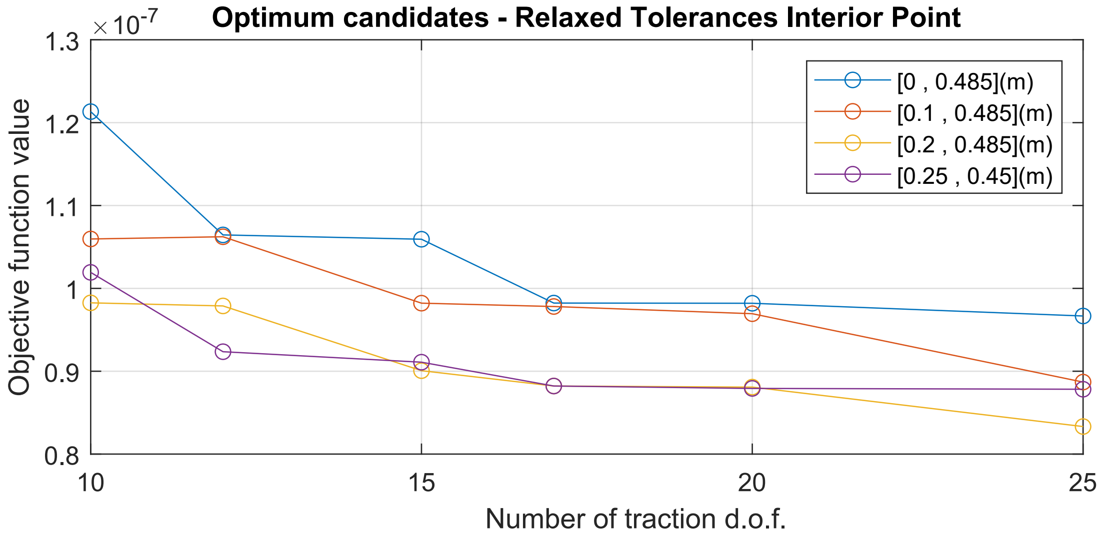


Figure 5.17: Values of the objective function for different number of degrees of freedom and different initial guesses for the *interior-point* algorithm with the tolerances relaxed.

As observed in Figure 5.17, when the design variables are restricted to a certain zone, and with the tolerances adjusted, the objective function values are lower. The exact location of the nonzero tractions is not known in the real experiment, and as it was shown in the previous semester project Viejo et al. [2019], the implementation of a routine which located the nonzero tractions affected the robustness of the whole program. This study indicates that a reduction of the objective function could be achieved, and more similar values for the traction d.o.f. However, as indicated in Figure 5.8, the fact of having the design variables enclosed in a small region, would increase the values of the condition number, and this is undesired.

Finally, in order to check the response of the inverse parameter tool, a Monte Carlo simulation is performed with the correct tolerances. Figures 5.18 and 5.19 correspond to the results obtained for 5 and 15 traction d.o.f. respectively. Purple lines correspond to the *original* problem, while blue lines correspond to the problem with the adjusted tolerances; the *benchmarked* one. The dashed lines capture the lower confidence intervals of the cohesive laws, while the dash-dotted lines are for the upper confidence intervals.

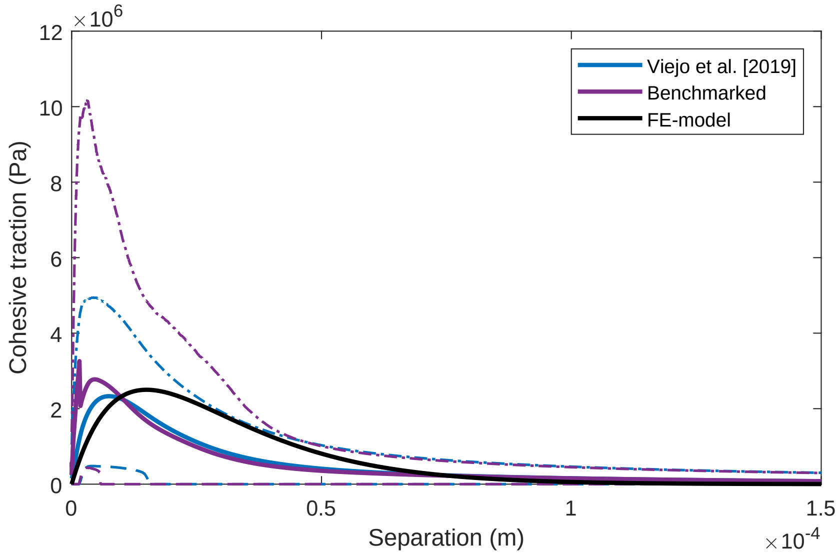


Figure 5.18: Resulting mean cohesive laws (continuous line) and confidence interval, delimited by the bounds (dashed and dash-dotted lines), obtained from the Monte Carlo algorithm using 5 traction d.o.f. Equally spaced traction peaks along the whole beam have been used as initial guess in the nonlinear optimizer.

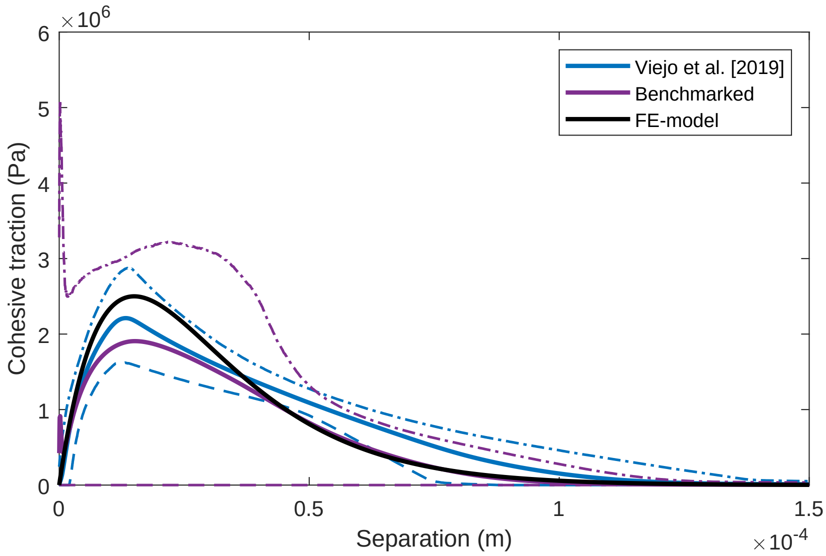


Figure 5.19: Resulting mean cohesive laws (continuous line) and confidence interval, delimited by the bounds (dashed and dash-dotted lines), obtained from the Monte Carlo algorithm using 10 traction d.o.f. Equally spaced traction peaks along the whole beam have been used as initial guess in the nonlinear optimizer.

5.2.3 Conclusions on the Nonlinear Optimization Study

The study performed in the previous paragraphs allows to evaluate the performance of the nonlinear optimizer on the inverse parameter identification tool. Firstly, it has been demonstrated that the problem at hand must be solved using the *interior-point* algorithm, since it shows a robust behaviour and a good trade-off between condition number and number of traction d.o.f. is always obtained.

The results stated in Table 5.3 show that the local minimum or minima can be found with the correct adjustment of the tolerances, as well as restricting the zone where the design variables are distributed. However, despite knowing the positive effects of the bounds on the design variables' placement, the latter one cannot be applied, as the location of the nonzero tractions can lead to potential errors of the inverse parameter identification tool. Besides the good performance of the algorithm on finding the optimum points, the shape of the interface tractions and cohesive law does not improve. Higher resemblance to the aimed tractions is not achieved, due to the formulation of the loading functions in the current inverse parameter identification routine. It is believed that the results describe a non-convex and highly irregular design space, wherein the algorithm falls into a local minimum.

Although the results obtained in Chapter 4 indicated a good performance, the nonlinear optimizer was wrongly applied, since the parameters discussed above, the tolerances, are not adjusted. Even though the MC results are better using the default parameters of the nonlinear optimizer, it is decided to maintain the reduced tolerances to ensure a consistent methodology which does not rely on default values and it is applicable for any case. Besides, the Monte Carlo results arise as an indicator of the high sensitivity of the current model with respect to the uncertainties of the input variables, as depicted in Figures 5.18 and 5.19.

As both the linear least-squares and the nonlinear optimization subroutine are proven to work properly, this is not reflected in the performance of the inverse parameter identification tool, which shows a lack of robustness when perturbations are present in the system. Therefore, it is believed that a new inverse parameter model, which can tackle the perturbations effectively, must be built.

6 | Changes Made to the Inverse Parameter Identification Model

In this chapter, the information gathered in Chapters 4 and 5 is used to build a new inverse parameter identification routine with improved performance. Three main improvements are presented: the identification of the force, the change in load functions and the introduction of linear constraints. These improvements are analyzed with a study of the impact of each on the linear least-squares system. After that, the performance of the new inverse parameter identification routine is evaluated with a MC simulation.

The results in Chapter 5 show that the inverse parameter identification routine provides a considerably wide confidence interval for the calculated cohesive law. The mathematical tools used to perform the inverse identification of the cohesive law, the linear and nonlinear optimization tools, have been proven to work properly. Thereafter, the excessive sensitivity of the routine to the uncertainty of the inputs is attributed to the beam-based load function formulation (see Appendix D). As a result, the changes performed on the inverse parameter identification algorithm are driven by the results obtained in Chapters 4 and 5.

Only the changes that mean an improvement are displayed here. A MC simulation is run after implementing each modification in MATLAB to prove its better performance.

The inverse parameter identification algorithm that is modified, is the routine developed in Viejo et al. [2019]. For an explanation of this beam-based model, see Appendix D.

6.1 Including the External Force as an Unknown of the Inverse Problem

In the previous semester project Viejo et al. [2019], the first step in the routine *InTraFiCa* is the subtraction of the displacements induced by the delamination force. This is done in order to have the displacement field only caused by the interface tractions. However, the calculation of the displacement field caused by the force is subjected to the uncertainty of four parameters: flexural rigidity (EI), shear rigidity (kAG), force value (F) and length or point of application (L). These parameters appear in Equation (6.1), which is the expression of Timoshenko beam theory for a cantilever beam [Timoshenko, 1930]. Moreover, the result of the correlation analyses in Chapter 4 show that EI and kAG have a big influence on the results.

It is believed that doing this subtraction, the error present in the force contribution is biasing the problem before the inverse parameter identification has even started. Therefore, it is decided to include the external force F in the inverse parameter identification procedure, calculating it in the same manner as the tractions.

$$v(x) = Ff(x) = F\left(\frac{x^2(3L-x)}{6EI} + \frac{x}{kAG}\right) \tag{6.1}$$

$f(x)$ is the compliance function, x is the longitudinal coordinate, being 0 at the clamp and L at the position of the force. The values of the compliance function are added as an extra column to the compliance matrix G. The new system of equations is shown in Equation (6.2).

$$\begin{bmatrix} G^L & G^F \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_j \\ \vdots \\ w_n \\ F \end{bmatrix} = \begin{bmatrix} G^L & \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_h \end{bmatrix} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_j \\ \vdots \\ w_n \\ F \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_h \end{bmatrix} \tag{6.2}$$

where n is the number of traction d.o.f. used to calculate the distributed load, h is the number of displacement data points of the centerline, f_i is the value of the force compliance at the same locations as the displacement measurement v_i, G^L is the compliance matrix developed in Viejo et al. [2019] and w_j are the unknowns of the linear least-squares, the traction d.o.f.

6.1.1 Effect of the Force Identification on the Least-Squares System

Following the same procedure as in Subsection 5.1.1, the singular values are analysed for different number of traction d.o.f. In this case, the focus is to study how the least-squares characteristics studied in Section 5.1.1 vary when the force applied at the end of the DCB is calculated by the algorithm. The singular values evolution with the increase in number of traction d.o.f. have a similar behaviour as the ones presented in Figure 5.1, gradually decaying to zero for every number of traction d.o.f. Therefore, the problem is still considered a discrete ill-posed problem.

The condition number progression against the number of traction d.o.f. can be seen in Figure 6.1.

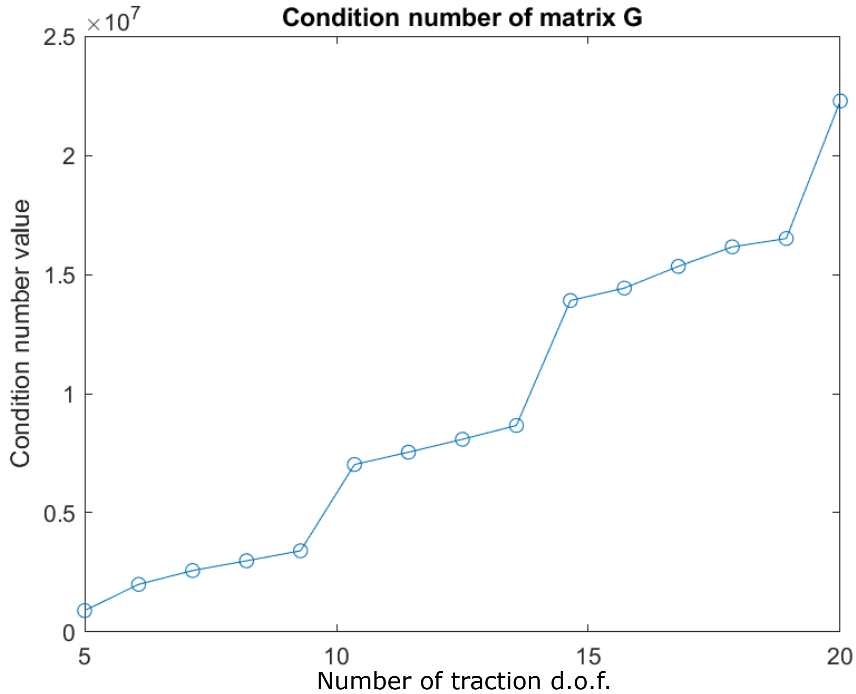


Figure 6.1: Condition number values for different number of traction d.o.f.

The condition number value in this case is of the order of magnitude of 10^7. When the force is not present in the compliance matrix G, the maximum condition number reached a value of an order of magnitude of 10^6 (see Figure 5.3). It can be seen in Figure 6.1 how the sensitivity to noise for 5 to 14 traction d.o.f. is an order of magnitude less (10^6) than for the rest of the cases (10^7). This indicates that between the solutions for different number of traction d.o.f., the sensitivity of the solution to input noise grows significantly above 14 traction d.o.f. used. This effect is clearly demonstrated in Figure 6.2, where the traction field calculated between 6 and 15 traction d.o.f. changes significantly when the noise is added to the input displacement.

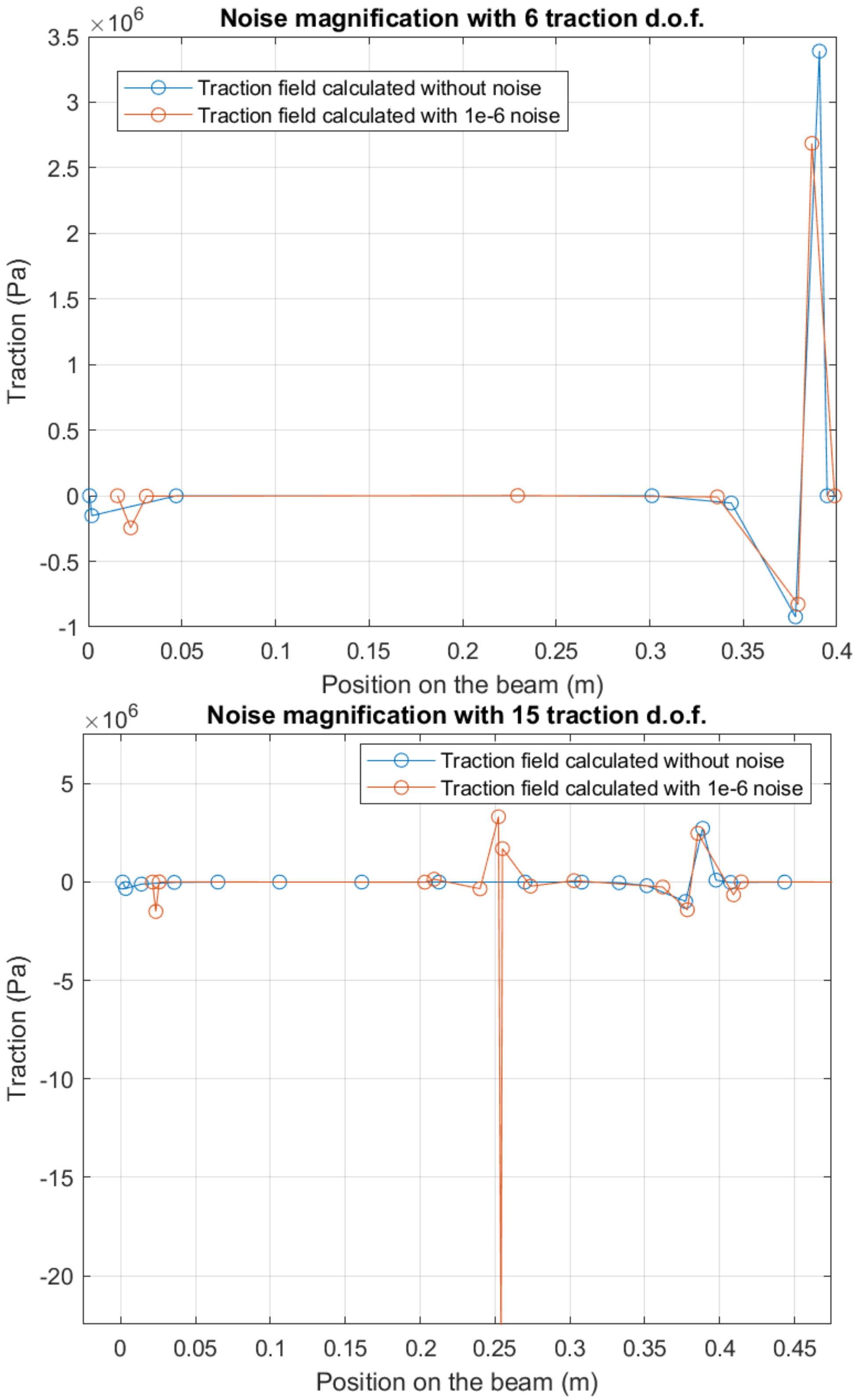


Figure 6.2: Difference between traction field solutions when noise is introduced in the input displacement using 6 (upper graph) and 15 (lower graph) traction d.o.f.

The effect of the noise affects the prediction of the onset traction independently of the number of traction d.o.f. that are used. The onset traction calculated when noise is added to the input displacement is been observed to be always under the onset traction calculated when there is no noise added to the input displacement (see Figure 6.2). Moreover, the effect of the noise is also seen on the oscillations in the area where the stress must remain equal to zero, when a high number of traction d.o.f. are used (see lower graph in Figure 6.2). Note that even though the noise effects can be clearly seen on both graphs in Figure 6.2, the amount of noise used for the study is an order of magnitude bigger than the one calculated for the DIC in Section 3.2.

The least-squares solution for the onset traction of the system where the force has to be predicted is bigger (almost 1 MPa) than the onset traction obtained with the system (see e.g. Figure 5.10) used in Chapter 5, where the force is not introduced. However, it has been proved that with the expected DIC noise magnitude (order of magnitude of 10^{-7}) the difference between onset tractions predicted is less than 0.1 MPa.

Even though the system loses robustness when having to predict the force that it is being applied to the DCB specimen, this formulation is still preferred than the one used previously in 5. That the system can predict the cohesive tractions and also the force applied at the end of the DCB specimen in a relatively accurate way is especially useful for the experimental part. It has been demonstrated, in the previous semester project Viejo et al. [2019] and it is shown in Figure 6.3, how a slight bad measurement of the force that corresponds to a DIC displacement can make the results to be completely wrong.

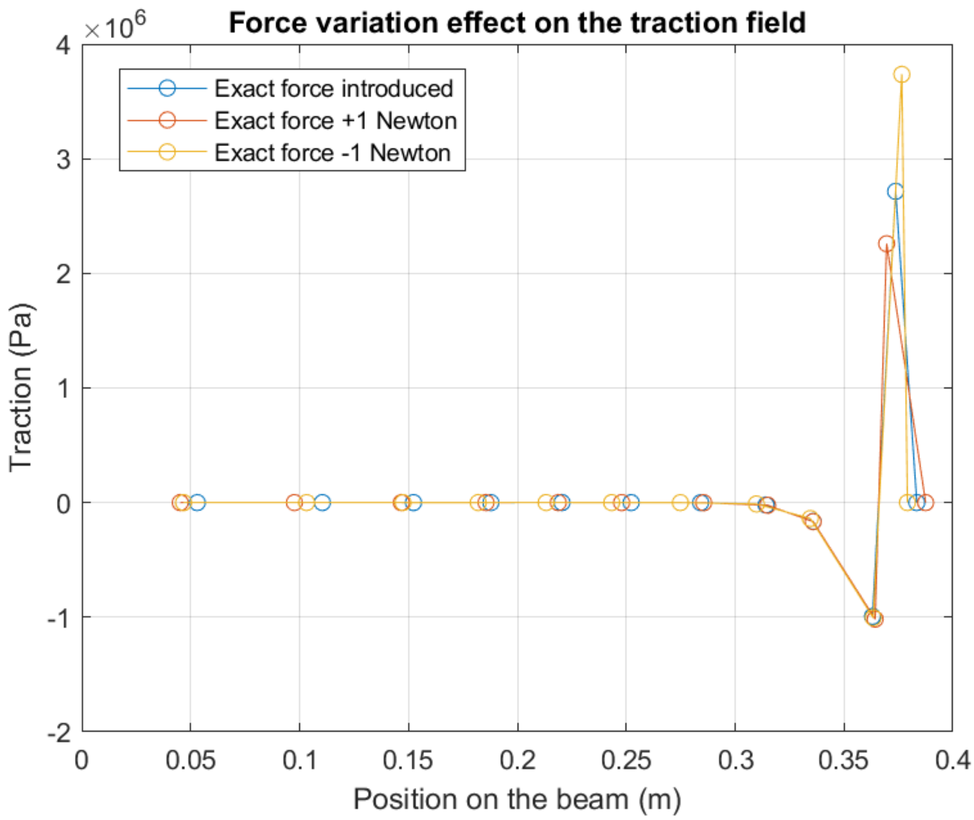


Figure 6.3: Difference in the solution when the force introduced to the system in Chapter 5 is not the exact force that causes the displacement field. 13 d.o.f. are used for the calculations.

Note that during the experiment, the variation of the force between 1 second along the entire delamination time is probably more (based on the previous semester project experience Viejo et al. [2019]) than $\pm 1N$ (which is the bound used in Figure 6.3). This means that, a perfect time synchronization between DIC photo and force measurement is essential for the system to work, which is fairly difficult to achieve in real life. This is why the least-squares system with the force prediction is preferred in this case, although it is more ill-conditioned than the one in Chapter 5.

6.1.2 Monte Carlo and Correlation Analysis

To assess the performance of the inverse parameter identification model including the force, a MC simulation is run. The results of the MC simulation are shown in Figures 6.4 and 6.5, for 5 and 10 traction d.o.f., respectively.

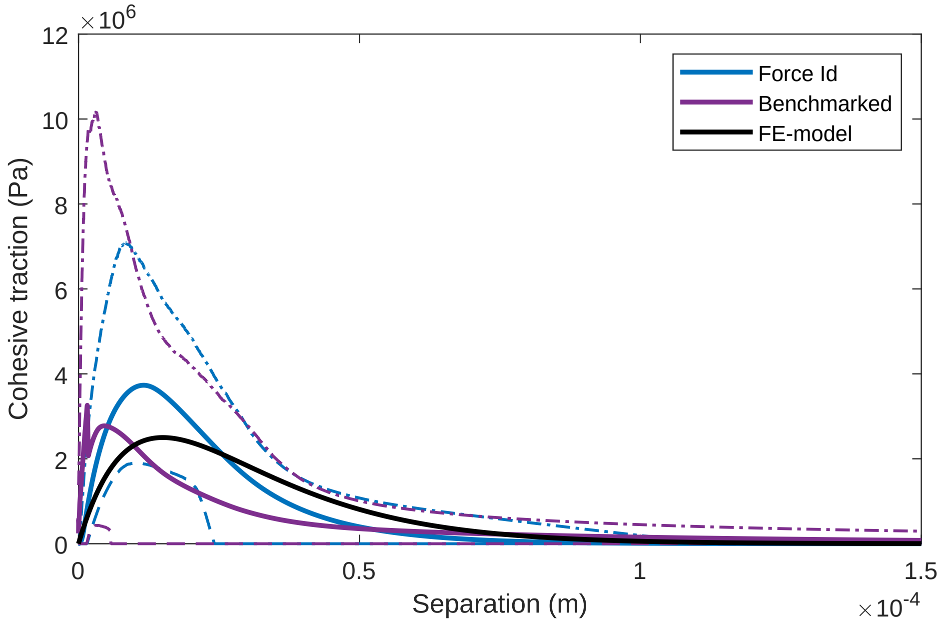


Figure 6.4: Resulting mean cohesive laws (continuous line) and confidence interval, delimited by the bounds (dashed lines), obtained from the MC algorithm using 5 traction d.o.f.. The *Benchmarked* curves refer to the results obtained after the benchmarking of the inverse parameter identification model, the curves termed *Force Id* are the routine including the force in the delamination and the curve labeled as *FE-model* is the one introduced as material property in the Finite Element model.

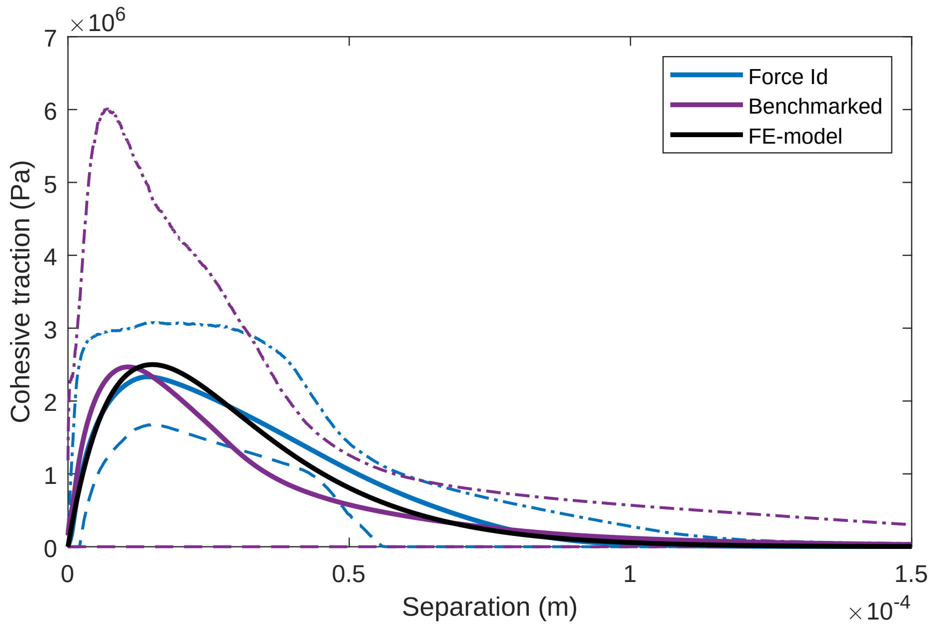


Figure 6.5: Resulting mean cohesive laws (continuous line) and confidence interval, delimited by the bounds (dashed lines), obtained from the MC algorithm using 10 traction d.o.f.. The *Benchmarked* curves refer to the results obtained after the benchmarking of the inverse parameter identification model, the curves termed *Force Id* are the routine including the force in the delamination and the curve labeled as *FE-model* is the one introduced as material property in the Finite Element model.

It can be seen how the inverse parameter identification routine that includes the force *Force Id* has a more robust performance with a mean curve closer to the *FE-model* curve and narrower confidence interval.

A correlation analysis is performed on the new model. The results from 5 and 10 traction d.o.f. are shown in Figures 6.6 and 6.7.

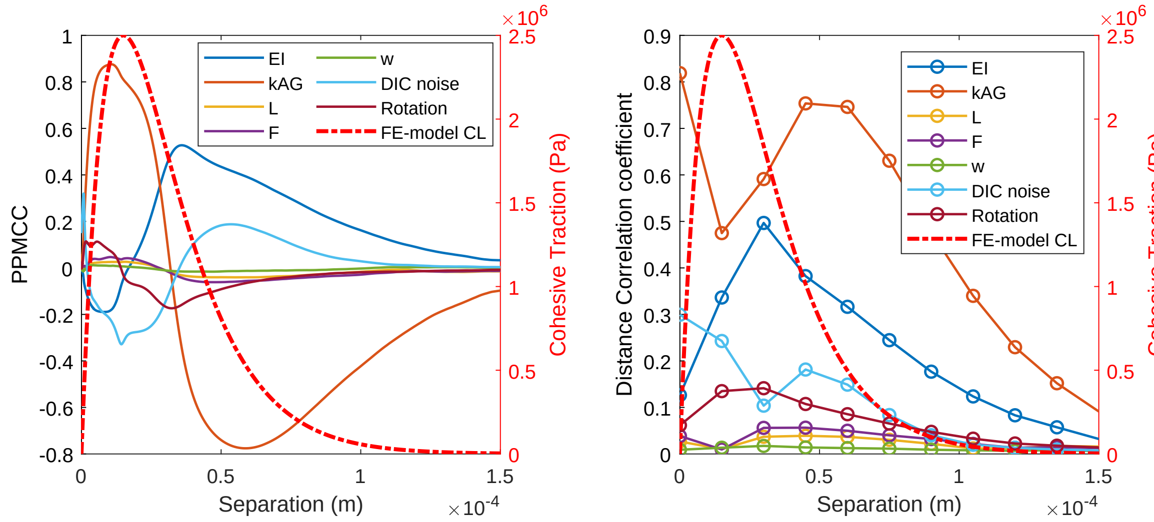


Figure 6.6: Correlation results when 5 traction d.o.f. are used in the inverse parameter identification routine using the model which calculates the force. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

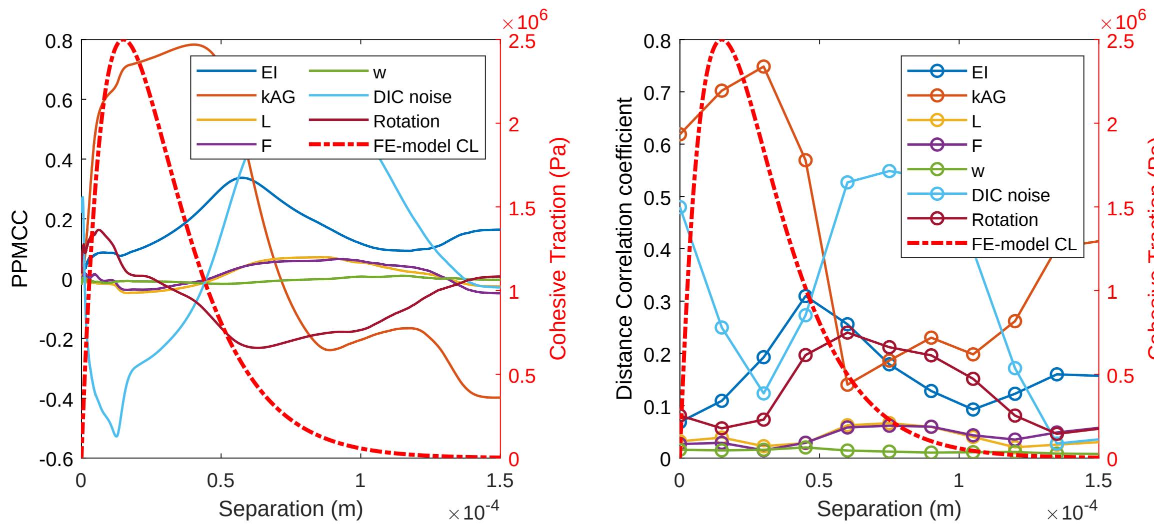


Figure 6.7: Correlation results when 10 traction d.o.f. are used in the inverse parameter identification routine using the model which calculates the force. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

In the results shown in Chapter 4, the parameter with most influence over the uncertainty of the results is the flexural rigidity EI. After changing the inverse parameter identification routine, the parameter driving the error is the shear rigidity kAG. The deformation caused by the force is dominated by the flexural rigidity EI, whereas for the displacements caused by the interfacial tractions, the shear rigidity kAG has a much higher influence. This result supports the idea that inspired this change: subtracting the contribution of the delamination force to the initial displacements before performing the inverse parameter identification of the tractions was introducing a big error to the final result.

The new inverse parameter identification model has a new output, the delamination force. This parameter can be used with validation purposes. In Figure 6.8, the distribution of force values calculated during the MC simulation using 10 traction d.o.f. in the inverse parameter identification is displayed.

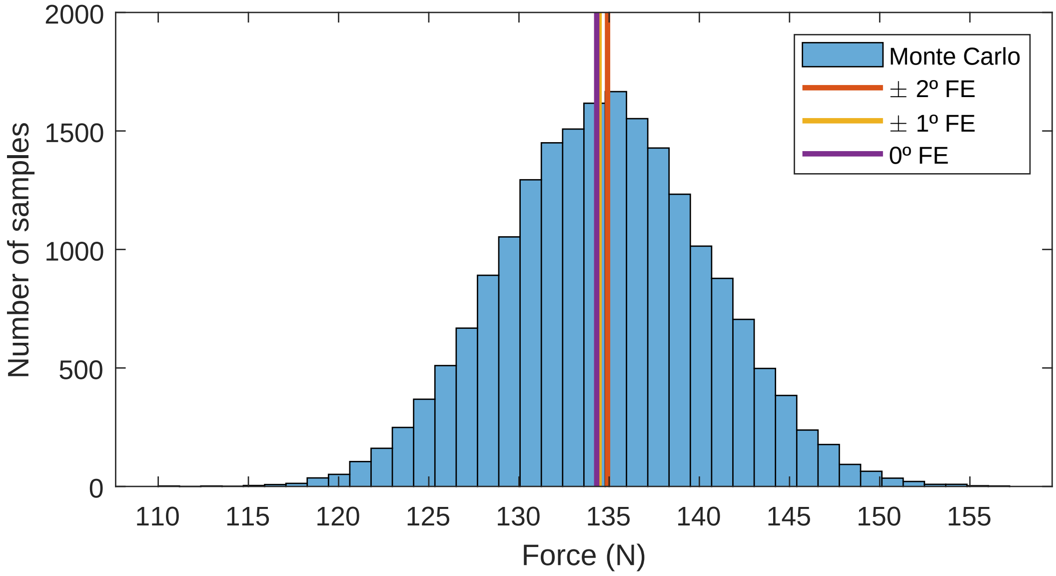


Figure 6.8: Distribution of forces values calculated by the inverse parameter identification routine. The vertical lines show the force value extracted form the FE-model for the different rotation values.

It can be seen how the distribution of obtained forces is centered on the FE-model values, which supports the validity of the inverse parameter identification model. The mean force values and the standard deviation of the MC results and the FE-model values are displayed in Table 6.1.

	0 deg	±1 deg	±2 deg	All rotations
FE-model	134.31 N	134.44 N	134.91 N	-
Monte Carlo mean	135.35 N	135.55 N	133.43 N	133.86 N
Monte Carlo standard deviation	5.74 N	5.78 N	5.61 N	5.76 N

Table 6.1: Values of the for different values of the rotation of the hinge from the FE-model and the MC simulation.

The impact of the uncertainties of the input on the calculated force values can be estimated by performing a correlation analysis. The results of correlating the force values with the input parameters is displayed in Table 6.2.

	EI	kAG	L	w	DIC noise	Hinges Rotation
PPMCC	0.9788	0.1427	-0.0561	0.0023	0.0100	0.0378
Distance Correlation	0.9687	0.1262	0.0512	0.0113	0.0124	0.0742

Table 6.2: Values of the correlation coefficients calculated from the values of the input parameters of the MC simulation and the calculated force.

As it could be expected, the material properties are strongly related to the calculated value of the force. This result can be used to validate the obtained cohesive law by comparing the maximum force recorded by the tensile test machine and the load calculated by the inverse parameter tool.

This change in the model has proven to have a positive impact in the robustness of the routine providing a narrower confidence interval for the cohesive law. Moreover, this change is translated into a simpler experimental procedure as only the data obtained by the DIC method is needed for the inverse parameter routine to calculate the cohesive law. As it was seen in the previous semester project Viejo et al. [2019], the synchronization of the measuring devices is a source of error, and this change avoids it.

6.2 Change in the Shape of the Load Function

The beam based-model developed in the previous semester project Viejo et al. [2019] makes use of triangular loads like the one shown in Figure 6.9.

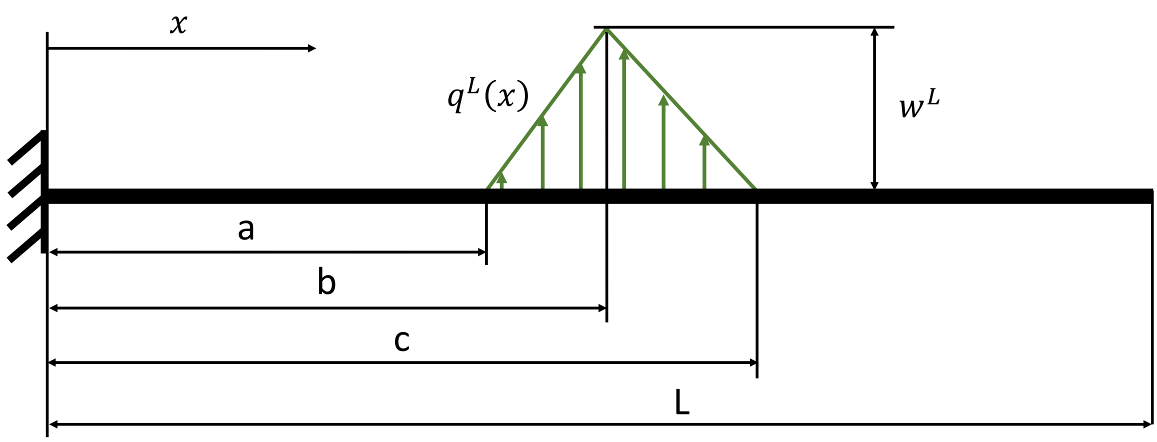


Figure 6.9: Triangular distributed load, used as elementary beam problem in the inverse parameter identification in Viejo et al. [2019]

The problem in Figure 6.9 is a beam clamped on the left and free on the rigth. The triangular distribution load distribution $q^L(x)$ is defined as

$$q^L(x) = w^L \begin{cases} \frac{x-a}{b-a} & \text{if} \quad a \leq x \leq b \\ \frac{x-b}{c-b} & \text{if} \quad b < x \leq c \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

where a, c and b are the positions of the ends and the peak of $q^L(x)$ respectively, as displayed in Figure 6.9, and w^L is the value of the distributed load at the peak b. The expression of the beam vertical displacement $v^L(x)$ caused by the load function in Equation (6.3) is Equation (D.6) in Appendix D.

In Chapter 4, it has been seen how the use a combination of triangular distributed loads to calculate the interfacial tractions requires from a high number of traction d.o.f. to provide an accurate representation. A high amount of traction d.o.f. represents a more ill-conditioned system which results in a calculation more sensitive to the input uncertainties, as it is shown in Chapter 5 for this problem. Therefore, a more suitable loading model is required for a higher accuracy and robustness in the determination of the cohesive law. As a solution, it is resolved to derive another analytic equation for a new load function, shown in Figure 6.10. This new shape has a part with curvature, which is assumed to allow for a better representation of the interfacial tractions with less d.o.f.

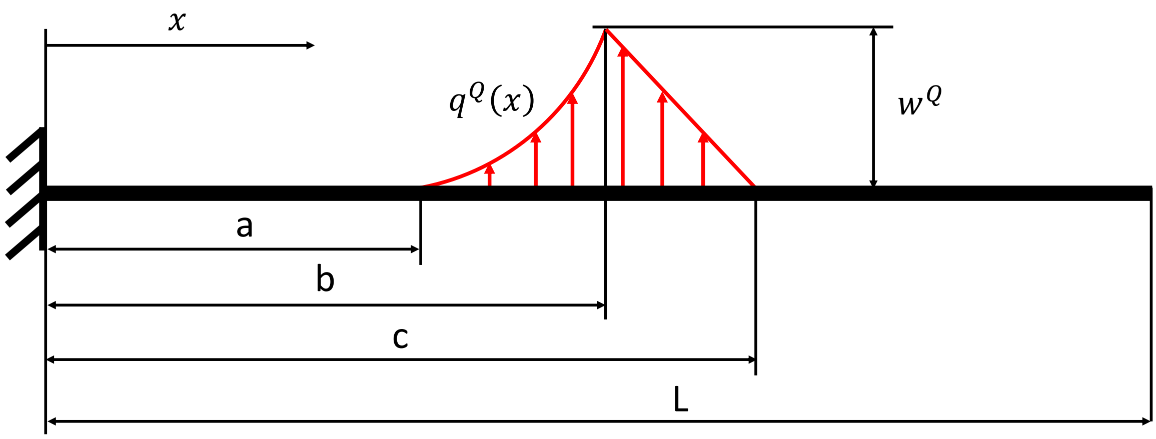


Figure 6.10: Quadratic load function used as elementary beam problem in the inverse parameter identification.

The distributed load $q^Q(x)$ is defined as

$$q^Q(x) = w^Q \begin{cases} \left(\frac{x-a}{b-a}\right)^2 & \text{if} \quad a \leq x \leq b \\ \frac{x-b}{c-b} & \text{if} \quad b < x \leq c \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

The expression of the displacement $q^Q(x)$ due to the distributed load in Equation (6.4), is derived following Timoshenko beam theory. It is displayed in Equation (6.5). This equation is obtained with the aid of a *MATLAB* script (see Appendix I) developed to derive the displacement equation for any distributed load.

$$v^Q(x) = w^Q \begin{cases} \begin{array}{c} \frac{x(a-b)}{4kAG} - \frac{x^3(b-c)}{12EI} - \frac{x^3(a-b)}{24EI} + \frac{x(b-c)}{2kAG} \\ + \frac{x^2(b-c)(2b+c)}{12EI} + \frac{x^2(a-b)(3a+12b)}{120EI} \end{array} & \text{if } x < a \\[2ex] \begin{array}{c} \frac{a^5-20a^3bx+10a^3x^2+30a^2b^2x-10a^2x^3-20ab^3x+5ax^4+5b^4x-x^5}{20kAG(a-b)^3} - \\ \frac{a^7-7a^6x+210a^3b^2x^2-140a^3bx^3+35a^3x^4-420a^2b^3x^2+210a^2b^2x^3-21a^2x^5}{840EI(a-b)^3} \\ + \frac{315ab^4x^2-140ab^3x^3+7ax^6-84b^5x^2+35b^4x^3-x^7}{840EI(a-b)^3} - \\ \frac{x^3(b-c)}{12EI} + \frac{x(b-c)}{2kAG} + \frac{x^2(b-c)(2b+c)}{12EI} \end{array} & \text{if } a \leq x \leq b \\[2ex] \begin{array}{c} - \frac{(a-b)\left(a^3+4a^2b-7xa^2+10ab^2-28xab+20b^3-70xb^2\right)}{840EI} \\ + \frac{-4b^5+5b^4c+15b^4x-20b^3cx+10c^3x^2-10c^2x^3+5cx^4-x^5}{120EI(b-c)} \\ \frac{a^2+3ab-4b^2}{20kAG} + \frac{2b^3-3b^2c+3c^2x-3cx^2+x^3}{6kAG(b-c)} \end{array} & \text{if } b < x \leq c \\[2ex] \begin{array}{c} \frac{a^2+3ab-4b^2}{20kAG} - \frac{(b-c)\left(4b^3+3b^2c-15xb^2+2bc^2-10xbc+c^3-5xc^2\right)}{120EI} \\ - \frac{(a-b)\left(a^3+4a^2b-7xa^2+10ab^2-28xab+20b^3-70xb^2\right)}{840EI} - \frac{-2b^2+bc+c^2}{6kAG} \end{array} & \text{if } c < x \end{cases} \tag{6.5}$$

It must be remarked that this new load function is not substituting the old one. The solution is found as a linear combination of the old shape, the triangular distributed load, and the new shape, the quadratic shape. As a result, the number of traction d.o.f. is the double, compared to the tractions model developed in the previous semester project Viejo et al. [2019]. The positioning of these loads is done in the same manner as in the previous semester project Viejo et al. [2019]. The triangular and quadratic loads are positioned in pairs, as shown in Figure 6.11, making the peaks and the ends of the individual distributions to match. If the *i*-th triangular load is defined by the locations a_i^L, b_i^L and c_i^L, and the *i*-th quadratic load is defined by the locations a_i^Q, b_i^Q and c_i^Q, their position is constrained as

$$a_i^L = a_i^Q = a_i \qquad b_i^L = b_i^Q = b_i \qquad c_i^L = c_i^Q = c_i \tag{6.6}$$

In addition, each pair of loads, $q_i^L(x)$ and $q_i^Q(x)$, is linked to the contiguous loads to its right, $q_{i+1}^L(x)$ and $q_{i+1}^Q(x)$, by setting the peak of the *i*-th distributed loads where the *(i+1)*-th loads have its initial point, see Figure 6.11. Additionally, each pair of loads, $q_i^L(x)$ and $q_i^Q(x)$, is also linked to the contiguous loads to its left, $q_{i-1}^L(x)$ and $q_{i-1}^Q(x)$, by setting the peak of the *i*-th distributed loads where the *(i-1)*-th loads have its end. This constraint follows the expression

$$a_{i+1} = b_i = c_{i-1} \tag{6.7}$$

This constraint was implemented for the triangular load functions in the previous semester project Viejo et al. [2019]. If the number of pairs of loads is n, i.e. there are n triangular loads $q_i^L(x)$ and n quadratic loads $q_i^Q(x)$, the loads are numbered being $q_1^L(x)$ and $q_1^Q(x)$ the ones closest to the clamp, and $q_n^L(x)$ and $q_n^Q(x)$ the ones closest to the free end. Consequently, the position of all the distributed loads can be expressed by a vector including the peaks of all the loads b_i, the end which is closest to the clamp, a_1, and the end which is closest to the free end, a_n, having then $n+2$ elements. This vector $\mathbf{x}$ is displayed in Equation (6.8).

$$\mathbf{x} = \begin{pmatrix} a_1 & b_1 & b_2 & \dots & b_j & \dots & b_{n-1} & b_n & c_n \end{pmatrix}^T = \begin{pmatrix} x_0 & x_1 & x_2 & \dots & x_j & \dots & x_{n-1} & x_n & x_{n+1} \end{pmatrix}^T \tag{6.8}$$

Using this notation, the peak of the i-th load functions, q_i^L and q_i^Q, is located at the point x_i.

To avoid numerical problems and nonphysical results, it is decided to set two constraints already implemented in the previous semester project Viejo et al. [2019] with the same purpose.

- The positions of the peak and the ends of the load must remain in order

$$a_i < b_i < c_i, \qquad \forall i \in [1, n] \tag{6.9}$$

- The tractions must remain in the beam domain.

$$a_i, b_i, c_i \in [0, L], \qquad \forall i \in [1, n] \tag{6.10}$$

An example of a combination of triangular and quadratic distributed loads is displayed in Figure 6.11.

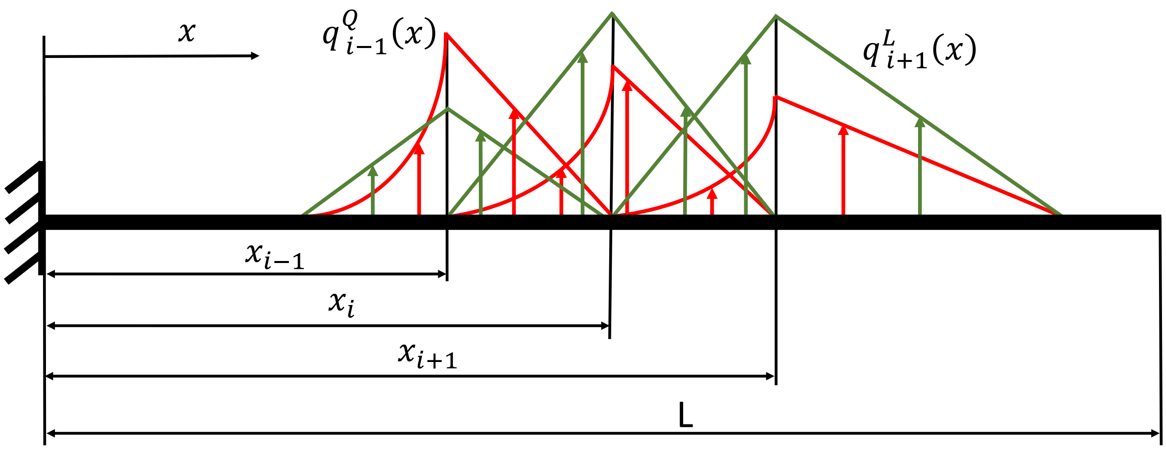


Figure 6.11: Example of three triangular (green) loads and three quadratic (red) loads.

To obtain a solution which is a combination of the triangular and the quadratic load distributions in the inverse parameter, the matrix G used to perform the pseudoinverse (see Appendix B) must be modified. Concatenating the compliance matrices, which relate the magnitude of a load with: the displacements of the triangular loads $G^L(\mathbf{x})$, the quadratic loads $G^Q(\mathbf{x})$ and the force G^F, the compliance matrix of the overall problem $G(\mathbf{x})$ is obtained. The construction of the compliance matrix from beam displacements for the inverse parameter identification of interfacial tractions is explained in Appendix D. The resulting system of linear algebraic equations is

$$\begin{bmatrix} G^L(\mathbf{x}) & G^Q(\mathbf{x}) & G^F \end{bmatrix} \begin{bmatrix} w_1^L & \dots & w_i^L & \dots & w_n^L & w_1^Q & \dots & w_i^Q & \dots & w_n^Q & F \end{bmatrix}^T = \begin{bmatrix} v_1 \\ \vdots \\ v_j \\ \vdots \\ v_h \end{bmatrix} \tag{6.11}$$

where v_j corresponds to the h vertical displacement measurements of the centerline. The system of algebraic equations in Equation (6.11) is ready to be solved using the linear least-squares, as explained in Appendix B. After calculating the magnitude of each individual load function w_i^L and w_i^Q, the final distributed load $q(x)$ is calculated as

$$q(x) = \sum_{i=1}^{n} \left(w_i^L q_i^L(x) + w_i^Q q_i^Q(x) \right) \tag{6.12}$$

6.3 Addition of Linear Constraints to the Linear Least-Squares Problem

The loading function model introducing the quadratic load functions has been proven to have an unstable response, as seen in Figure 6.12, where solutions are numerically correct but do not represent the desired behaviour observed in real life.

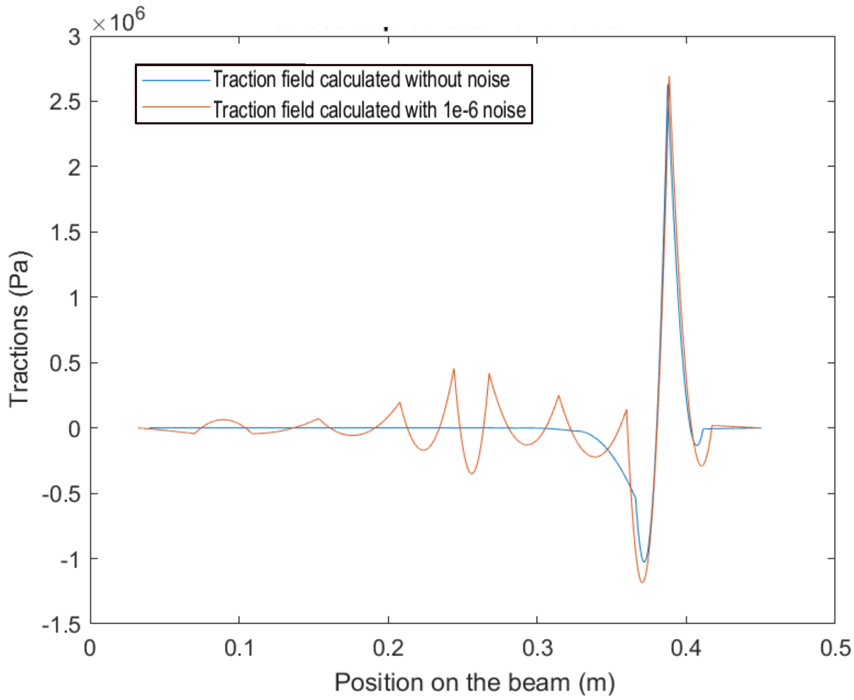


Figure 6.12: Noise effect on the solution variance due to the ill-conditioning of the coefficient matrix. Results shown are obtained with 20 d.o.f.

The solution space contains distributions which are known to be far away from the interfacial tractions (like the one shown in Figure 6.12). To overcome this problem, linear equality constraints are introduced to the linear least-squares, see Section B.3.2 in Appendix B, still having a closed form solution available. Three physical constraints are introduced to the resulting distributed load to restrain the solution space:

- **C^1-continuity:** enforcing the continuity of the first derivative of the loading functions along the longitudinal dimension of the beam.
- **Force equilibrium:** imposing a sum of forces in the vertical direction equal to zero.
- **Moment equilibrium:** make the sum of moments generated by the delamination and the distributed load be zero.

In addition, two more constraints are implemented as a result of the knowledge of the interface traction distribution of the DCB. The normal stress at the delamination interface of the lateral face of the DCB FE-model, see Chapter 2, is shown in Figure 6.13. The two red circles point out the ends of the area where the the stress is nonzero. It can be seen how it ends with zero slope $\partial\sigma/\partial x = 0$ at both sides.

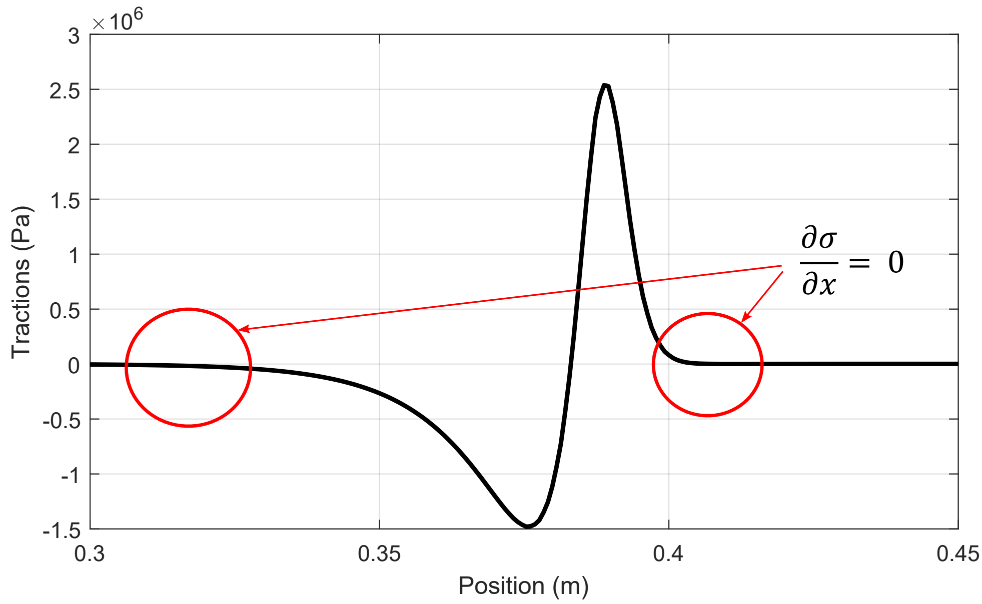


Figure 6.13: Interfacial traction distribution of the DCB specimen in the FE-model.

As a result, it is decided to enforce the behaviour shown in Figure 6.13 in the solution of the inverse parameter identification.

- **Zero slope** at the point which is closest to the clamp a_1.
- **Zero slope** at the point which is closest to the free end c_n.

The creation of a matrix Z of linear constraints is explained, to express the constraints as

$$Z \left[w_1^L \dots w_i^L \dots w_n^L \; w_1^Q \dots w_i^Q \dots w_n^Q \; F \right]^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.13}$$

where $Z \in \mathbb{R}^{p \times 2n+1}$, being p the number of constraint equations and n the number of pairs of triangular and quadratic distributed loads. The constraints have to be expressed in terms of the magnitude of the traction peaks, w_i^L and w_i^Q, and the force F.

The derivation of the constraints to obtain the matrix $Z \in \mathbb{R}^{p \times 2n+1}$ is displayed in Appendix H.

6.3.1 Effect of the Constraints on the Linear Least-Squares

With the implementation of the constraints, the current problem now has two matrices, the coefficient matrix G and the constraint matrix Z (following the notation used in Section B.3.2 in Appendix B). Therefore, the ill-conditioning has to be studied again, making use of the GSVD introduced in Section B.3.2. Thanks to the GSVD of the matrix pair (G, Z), the singular values θ_i, μ_i for both matrices G and Z are calculated, to obtain the generalized singular values γ_i using equation (B.13).

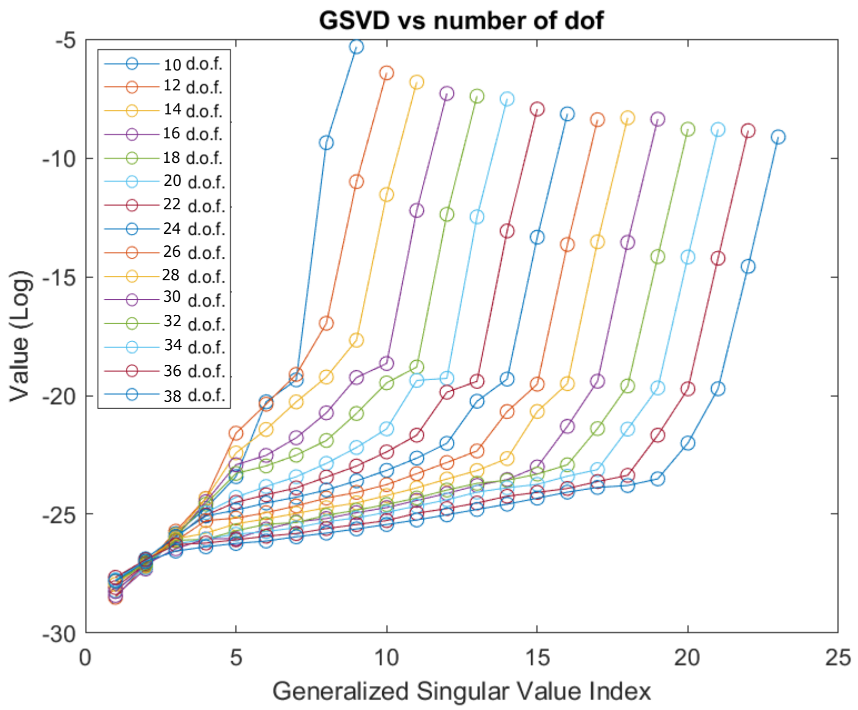


Figure 6.14: Generalized Singular Values γ_i of the matrix pair (G, Z) for different number of traction d.o.f. (from 5 to 19).

It can be seen in Figure 6.14 that the decomposition has inherited the ill-conditioning from the coefficient matrix G. This is known because the constraint matrix Z is perfectly conditioned (its condition number is equal to one and singular values $\mu_i = 1$), so the ill-conditioning seen in Figure 6.14 has to be produced purely by matrix G. Moreover, orthonormal vectors u_i and v_i also show the oscillatory behaviour characteristic of the discrete ill-posed problems (similar to the one shown in Figure 5.2). With the introduction of the constraint matrix, the condition number to be evaluated is the one from matrix X, as explained in Section B.3.4 in Appendix B.

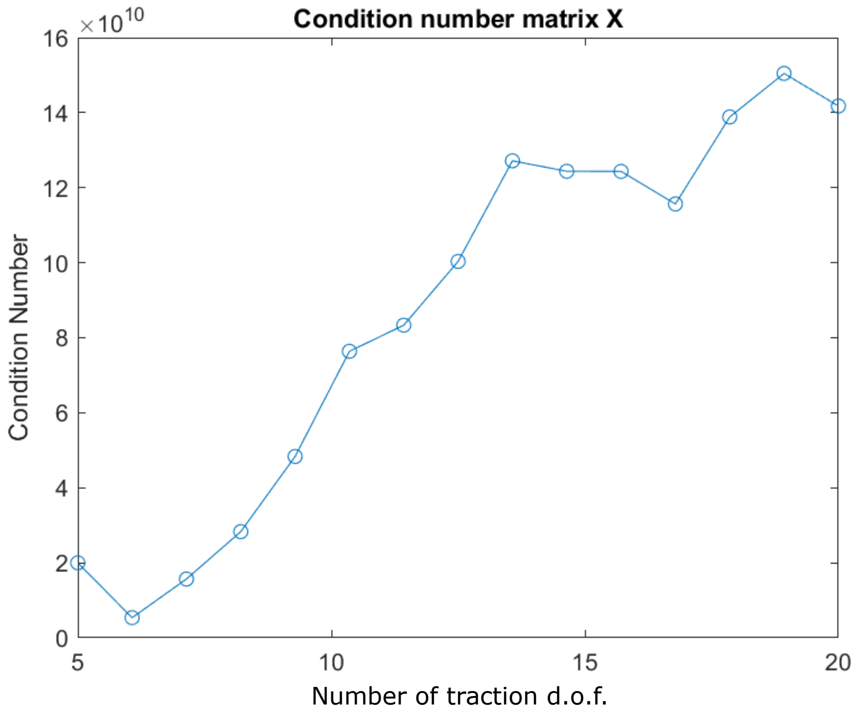


Figure 6.15: Condition number of matrix X for different number of d.o.f. (from 5 to 20).

Analysing the condition number of the matrix X depending on the number of d.o.f., shown in Figure 6.15, it can be seen that for the new formulation there is a relatively big increase on the instability of the system when a new d.o.f is added to the solution.

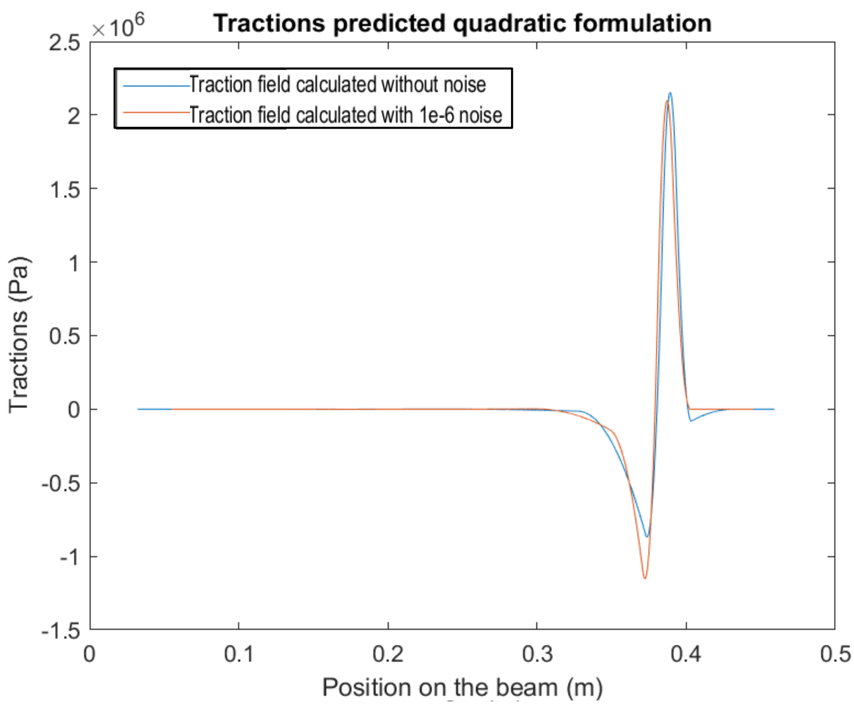


Figure 6.16: Solution of the quadratic triangular load function formulation with constraints with noise (orange line) and without noise (blue line) for 10 d.o.f.

Figure 6.16 shows that the imposition of constraints plus the solution of the system with GSVD makes the response much more stable to the noise effect, almost to the point of being negligible (comparing Figures 6.12 and 6.16). This is due to the fact that the system is solved as a weighted least squared problem, which is mentioned in Section B.3.3. The problem is solved as a Weighted Least-Squares, with larger weights put on the constraint matrix Z, which makes the solution to be more driven by the constraint equations than the noise component present in the observations vector b.

Besides, following the same procedure as in Section 5.2.2, the influence of different initial guesses onto the final distribution is checked. Figure 6.17 shows that the obtained traction distributions differ depending on the position where the d.o.f. are initially distributed. It is also observed that the objective function values are not that similar as the ones obtained using the inverse parameter identification routine of the previous semester project Viejo et al. [2019], see Table 5.3, which is attributed to the algorithm falling into many different local minima. It is believed, that this happens due to the non-convexity of the design space. Unfortunately, the study of this effect with the new inverse parameter identification model is left for further work.

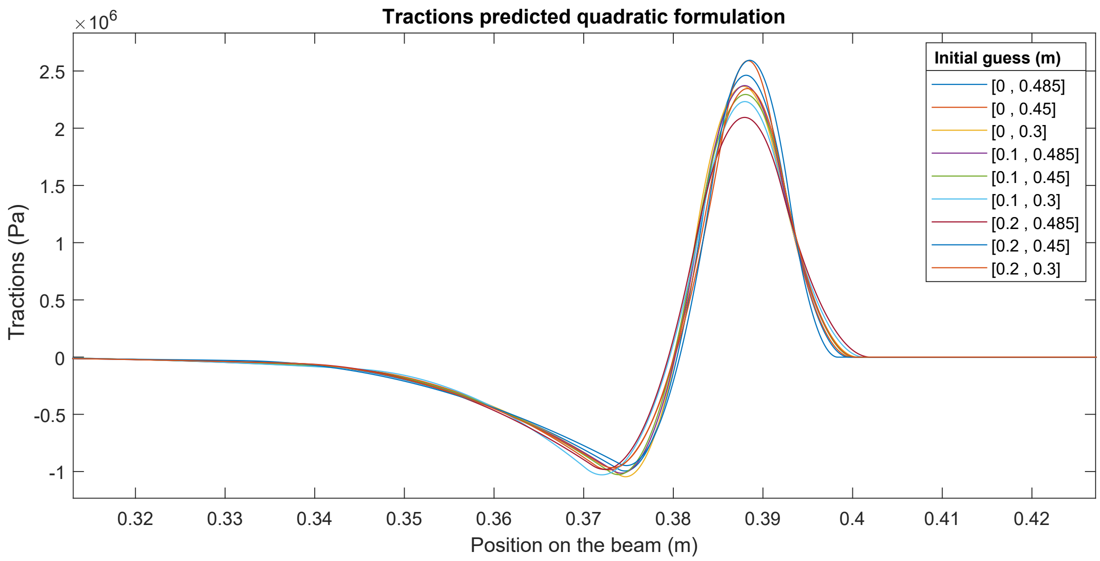


Figure 6.17: Solution of the quadratic formulation with constraints with different initial guesses, for 20 traction d.o.f.

6.4 Performance of the New Inverse Parameter Identification Model

To confirm the conclusion obtained from the analysis of the linear system, a MC analysis is performed over the developed model. The results are shown in Figures 6.18 and 6.19.

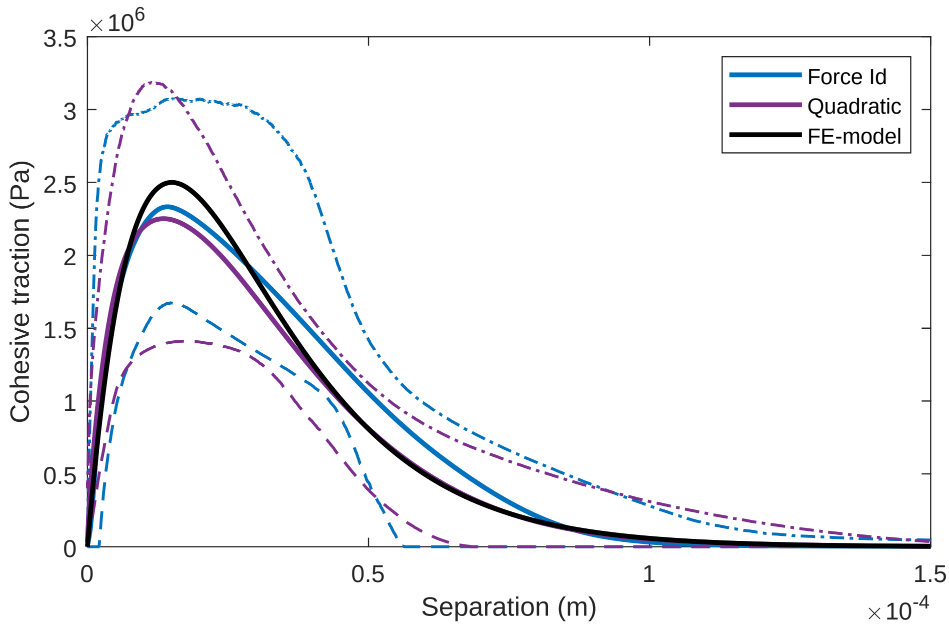


Figure 6.18: MC results of the inverse parameter identification routine which calculates the force *Force Id* and the constrained model with quadratic load functions *Quadratic* when 10 traction d.o.f. The curve *FE-model* corresponds to the cohesive law introduced in the FE-model as a material property. The continuous lines correspond to the mean cohesive law and the dashed lines correspond to the bounds of the confidence interval at 95% of confidence level.

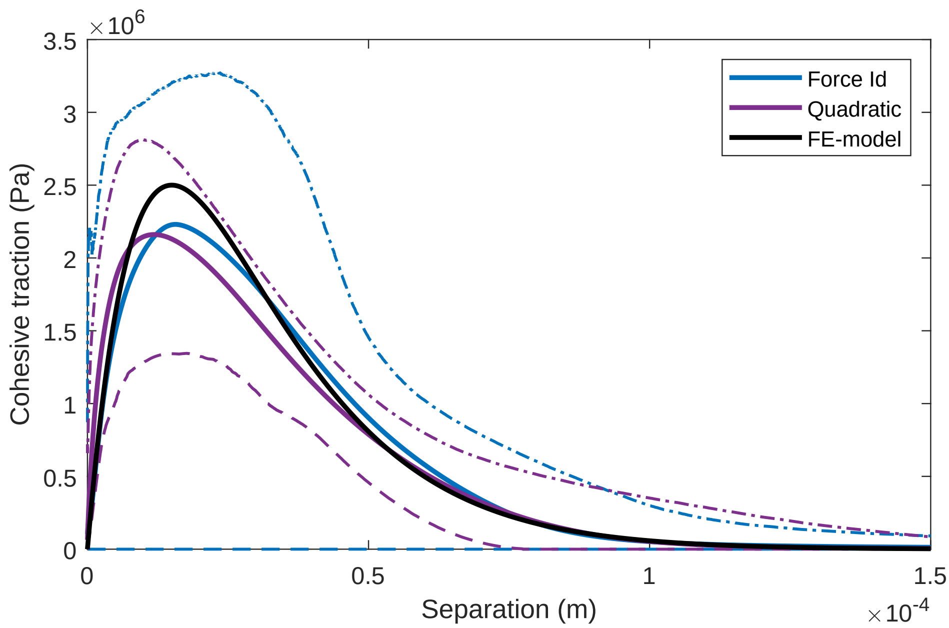


Figure 6.19: MC results of the inverse parameter identification routine which calculates the force *Force Id* and the constrained model with quadratic load functions *Quadratic* when 20 traction d.o.f. are used. The curve *FE-model* corresponds to the cohesive law introduced in the FE-model as material property. The continuous lines correspond to the mean cohesive law and the dashed lines correspond to the bounds of the confidence interval at 95% of confidence level.

It can be clearly seen how the new inverse parameter identification model provides more accurate results. The mean curves are both very close to the *FE-model* curve, however, the confidence interval of the *Quadratic* is significantly narrower.

A correlation analysis is performed on the MC results to identify the parameters driving the error. The results are shown in Figures 6.20 and 6.21.

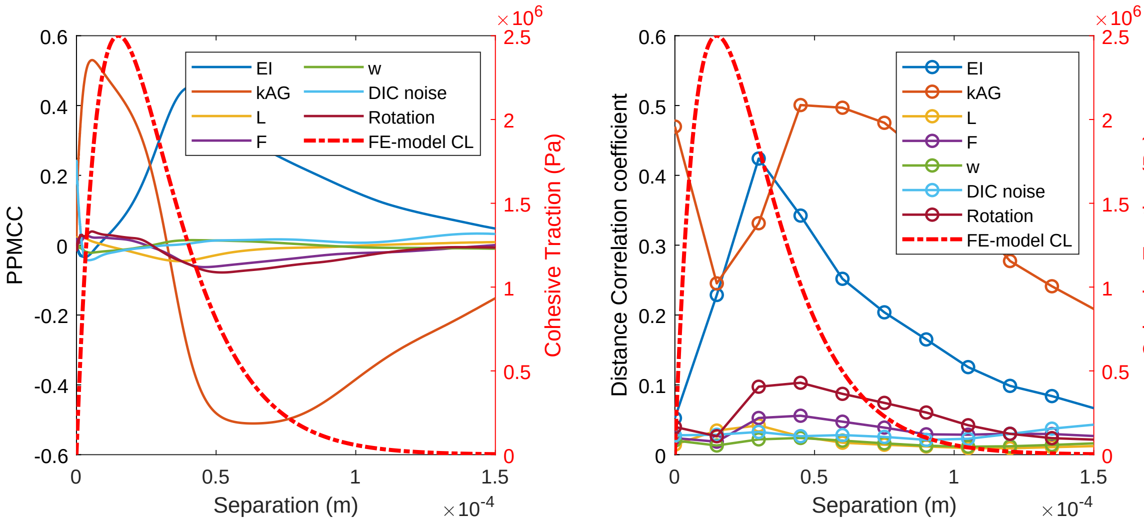


Figure 6.20: Results of the correlation analysis of the MC results when 10 traction d.o.f. are used in the inverse parameter identification. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

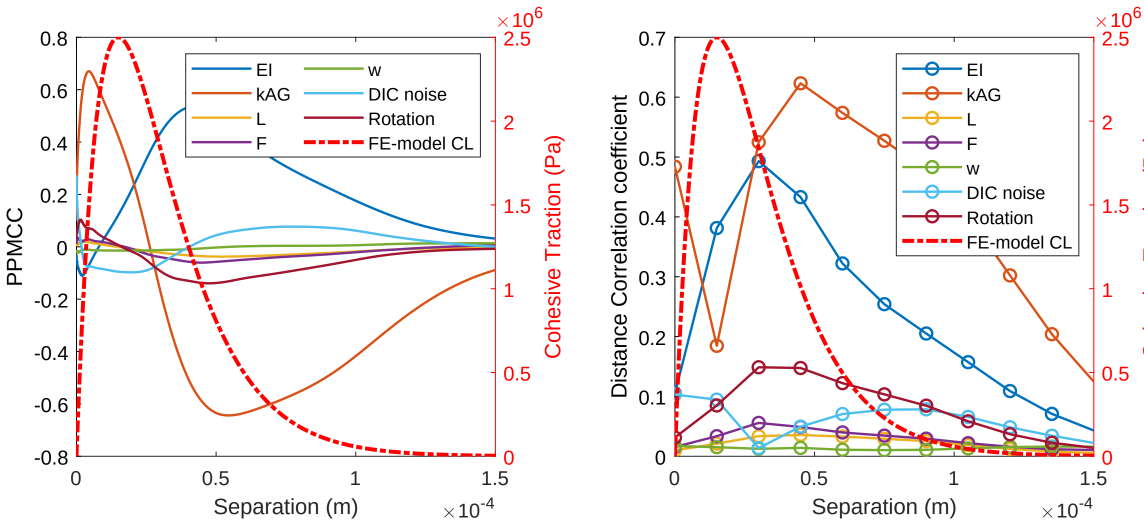


Figure 6.21: Results of the correlation analysis of the MC results when 20 traction d.o.f. are used in the inverse parameter identification. The dash-dotted red curve corresponds to the FE-cohesive law. The vertical axis of the cohesive law is the one on the right.

The most important parameters are the shear rigidity *kAG* and the flexural rigidity *EI*. The correlation analysis show the same behaviour as the beam-based model with triangles which includes the force in the calculation.

6.5 Conclusion of the model changes

Three changes have been introduced to the inverse parameter identification routine: including the force in the calculation, changing the shape of the load functions and adding linear constraints to the linear least-squares system. This changes have been proven to have a positive effect on the robustness of the algorithm providing a narrower confidence interval for the cohesive law.

The inverse parameter model with constraints outperforms any other model shown here. However, this result might be related to the shape of the cohesive law chosen for the FE-model, an exponential law (see Appendix E). The shape of the quadratic load functions is more suitable for the prediction of this particular type of cohesive law. Consequently, the inverse parameter identification model which uses triangular load functions and includes the force in the calculation is also going to be used to analyze the experimental measurements in Chapter 8.

7 Experimental Procedure

This chapter explains the steps taken during the experimental part of the project. First, the design considerations regarding the experimental procedure are described and the setup is proposed. The specimens used for the experiment are described. The procedure followed during the experimental activity is explained.

7.1 Design of the Experiment

All the analysis done throughout all the report so far are done in order to be able to design a proper experimental procedure that can be used effectively with the inverse parameter identification algorithm proposed. Therefore, a summary of the conclusions (concerning the experiment) obtained so far is done.

7.1.1 Experimental Considerations from the Third Semester Project

As mentioned in Section 1.4.1:

- The DIC method has proven an accurate method for measuring displacement fields. However, it is very dependent on the speckle pattern quality and quality of the image itself (focus of the camera on the surface).
- Self-similarity of the crack is not a necessity for the proposed algorithm. Even if the crack growth is not stable at all times and the FPZ size changes, the traction field predicted is able to adapt to it and have different shapes and values.
- Involving too many pieces of equipment that must be synchronized in time and measuring a lot of variables is most of the times not accurate enough and can cause big errors on the results.
- The moment rig machine is a piece of equipment that is not easy to use, and that requires a complex design and manufacturing processes. So it is not a machine that is available in every laboratory.

From these statements it is first clear that DIC is a reasonable experimental method to measure continuous displacement fields accurately. Nonetheless, a good quality speckle pattern must be obtained. Moreover, picture quality has to be acceptable, for which a camera calibration procedure is suggested in order to avoid this problem. On the other hand, the moment rig machine has many different reasons that advocate for its substitution in favour of a more simple, accessible machine (a simple tensile test machine).

7.1.2 Experimental Considerations from the Anticlastic Bending Study and the Hinges Rotation Study

From Chapter 2, the following conclusions are obtained:

- The influence of the anticlastic bending on the studied specimen does not suppose a big difference in the results obtained from the inverse parameter identification algorithm in the neutral beam is taken as input displacement field.
- The effect of the misplacement of the hinges on the displacement does not affect considerably the results when the rotation is below 2º. This is a rotation that can be seen by the human eye, so there is no need of suggesting a precision drilling to align the holes for a perfect positioning of the hinges.

Therefore, the DIC measurements are to be taken at the neutral line of the lateral face for correct results and the position of the hinges is not an important factor to consider when done properly.

7.1.3 Experimental Considerations Extracted from the Monte Carlo Analysis and the Force Prediction

From Chapter 6, information related to the experiment includes:

- Shear rigidity first and flexural rigidity second, are the two parameters that are considered most important according to the findings obtained with the Monte Carlo method.
- When the number of d.o.f. used is higher then the noise of the DIC becomes an important parameter to influence the results.
- The force applied to the end of the DCB specimen is also predicted by the inverse parameter identification algorithm.

With these statements, the importance of doing an experimental procedure prior to the delamination experiment to measure accurately the real material properties of the specimen becomes clear. Moreover, the introduction of the force on the inverse parameter algorithm indicates that it does not have to be measured in the experiment anymore, which eliminates the necessity of time synchronization between equipment.

7.2 Experiment Proposed

All the information gathered in the previous section is used to design an experimental procedure to be performed in real-life. First of all some preliminary procedures must be done:

1. **Three-point bending:** this experiment is used to determine both shear and flexural rigidity properties of the specimens later used for delamination. These quantities are the input of the inverse parameter algorithm.
2. **DIC-camera calibration:** this procedure is used to achieve good quality pictures and be able to eliminate lens distortion or compensate for out of plane movement/camera rotation [Jones and Iadicola, 2018].
3. **Geometric measurements of the specimen:** these values are also going to be used as input for the inverse parameter identification

The setup proposed for the experiment is described in Figure

The steps to be followed for the actual execution of the experiment are:

1. Paint the specimens with the speckle pattern and check for its quality.
2. Place the hinges on the ends of the DCB specimen.
3. Place the specimen on the tensile machine.
4. Place the camera at a reasonable distance from the specimen and check its perpendicularity to the lateral face of the specimen.
5. Place the lamps and check for the lightning.
6. Take undeformed pictures of the specimen.
7. Select a proper opening rate on the tensile machine and start the test.
8. Start the image acquisition process when the crack starts travelling.
9. Finish the experiment and post-process the results.

A more detailed explanation of the procedure, equipment, values and measurements is given in the following sections.

7.3 Specimen Data

First of all, note that the specimens used for the experiment are different from the ones that were meant to be used for this study (see 3) (regarding geometry, properties and probably cohesive law behaviour) due to external circumstances. Therefore the specimens used have the same geometry and properties as the ones used on the previous semester investigation Viejo et al. [2019], which properties cannot be shown due to privacy.

The specimens that are going to be used for the experimental procedure are made of unidirectional (UD) glass fiber composites bonded in epoxy resin. The specimens have a pre-crack made with a teflon insertion before the lay-up process. Five specimens are going to be used for the analysis, which geometrical properties are presented in Table 7.1.

Specimen No.	**Length**	**Width**	**Thickness**	**Inital crack length**
-	(mm)	(mm)	(mm)	(mm)
1	700	27,9	8,5	30
2	700	28,3	8,5	30

Table 7.1: Table showing the geometric properties of the DCB specimen

All the specimens have to be prepared accordingly with the experimental procedure. Therefore, holes are made in the specimen with a regular drill and then the hinges are placed on top of the holes. Moreover, due to the fact that a DIC technique is going to be used, the specimens must be painted with black and white paint to conform a speckle pattern (see Figure 7.6).

7.4 DIC Procedure

The method chosen to measure the displacement field, used as input in the inverse parameter model, is done with a DIC analysis of the delamination process. DIC -also denominated white light speckle technique- is an optical-numerical full-measuring approach that determines in plane displacements at the surface of an object under any loading situation [Lecompte et al., 2006]. Coordinate fields are measured and used for the calculation of the needed parameters, which can be displacements, strains or stresses [Jones and Iadicola, 2018]. It is a highly versatile method that its widely applied in the deformation characterization for different types of materials and with different object sizes.

The proposed setup, shown in Figure 7.1, is based on 2D-DIC with a camera and two light sources to illuminate the specimen. The equipment used is a 8 bit, 2048x2048 ARAMIS 4M camera system, with a 50 mm lens of family C (According to the ARAMIS Hardware manual). To perform a 2D-DIC analysis the surface of the specimen is assumed to be planar for the whole delamination process and to remain perpendicular to the camera axis [Jones and Iadicola, 2018]. Moreover, additional control must be made if out-of-plane movement is observed (specimen buckling, rotation or translations), that might cause errors in the DIC (which is not the case for this project) [Jones and Iadicola, 2018].

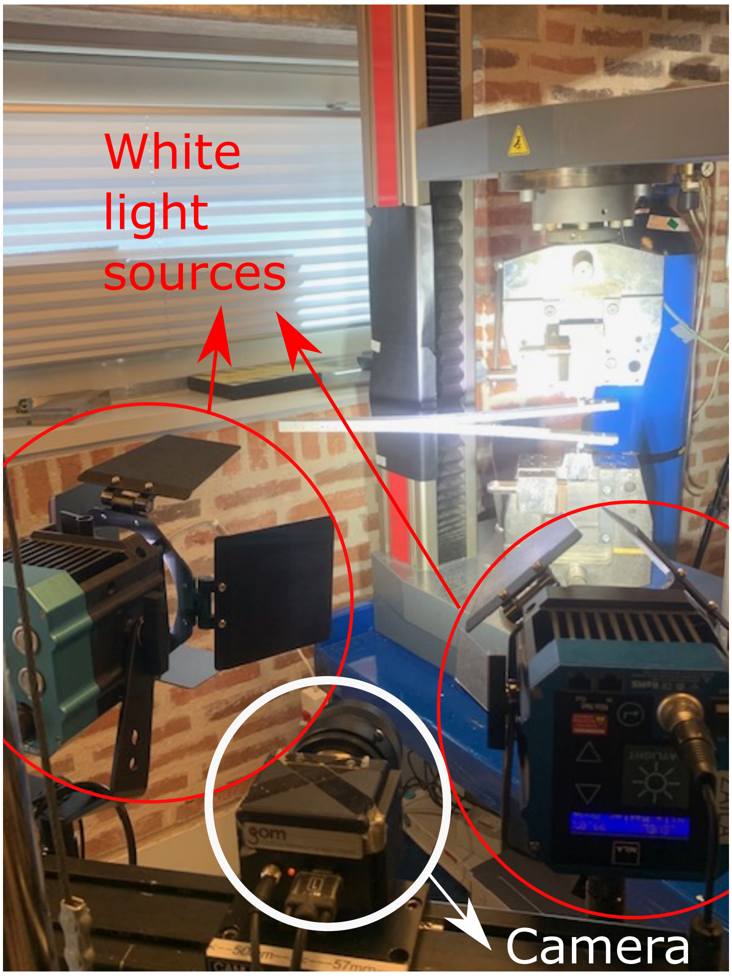


Figure 7.1: Image showing the DIC equipment (camera and light sources) position during a delamination experiment.

7.4.1 Speckle Pattern Analysis

For an optimal use of the method, the specimen to be used has to be covered with an speckle pattern of roughly circular black "speckles" of (preferably) uniform size and randomly located across the surface. The pattern features depend on the characteristics of the images to be analysed. Too small size of the circles (smaller than 3 pixels) may be aliased and give errors in the correlation. On the other hand, too big circles require a higher subset size, but can be more accurate than small speckles (as demonstrated in Lecompte et al. [2006]). However, a bigger size of "speckles" is preferred as they do not affect the results negatively.

The coverage of the speckle is also an important feature to take into account. For an area covered with circular dots a proportion of 20/40% of pattern density is desired [Jones and Iadicola, 2018]. It has been shown in Carter et al. [2014] how a too low density of the pattern have significant impact on the signal noise and give errors in the calculations.

For this project, the control over the speckle pattern characteristics is limited, as the paint is applied using a spray. For checking the contrast of the speckle pattern achieved, the greyscale distribution of the region of interest for the pattern of specimen is analysed using a histogram (see Figure 7.2).

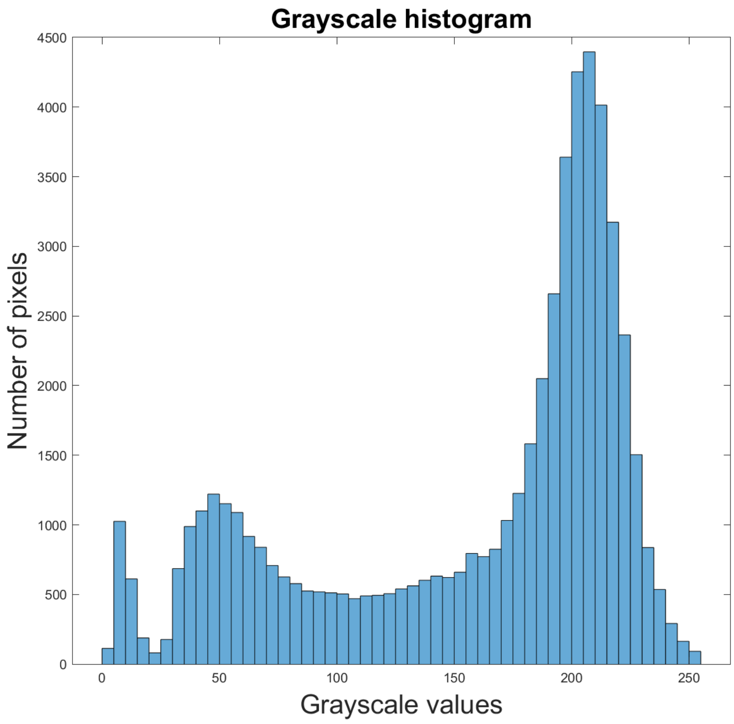


Figure 7.2: Picture of speckle pattern of a part fo the ROI from the specimen used.

The ideal speckle pattern should contain high contrast between black and white [Jones and Iadicola, 2018], which is the case for the speckle pattern shown in the figure above. There is one peak at a low greyscale value (black) and another one on the high greyscale value (white).

7.5 Tensile Test Machine

The device which is used to deliver the force to the DCB specimen in the mode I delamination experiment is the Zwick/Roell Z100. From now on denominated Zwick. The Zwick machine is located at the Materials and Production Engineering Lab at Aalborg University. It is a servo-controlled testing instrument, which consists of a fixed rig and a mobile crosshead. The crosshead can move upwards and downwards. Therefore, both tensile and compressive forces can be achieved depending on the objective of the experiment. Figure 7.3 shows the current configuration of the Zwick, wherein the mobile crosshead and the fixed rig can be spotted. Additionally, the clamping system can be seen, which consists of two headers. One attached to the rig, and the other moving together with the crosshead. The clamps are attached to the machine by a set of bolts. If it is needed to rotate the headers, it can be done by unscrewing the bolts and then fasten the system again in the desired configuration.

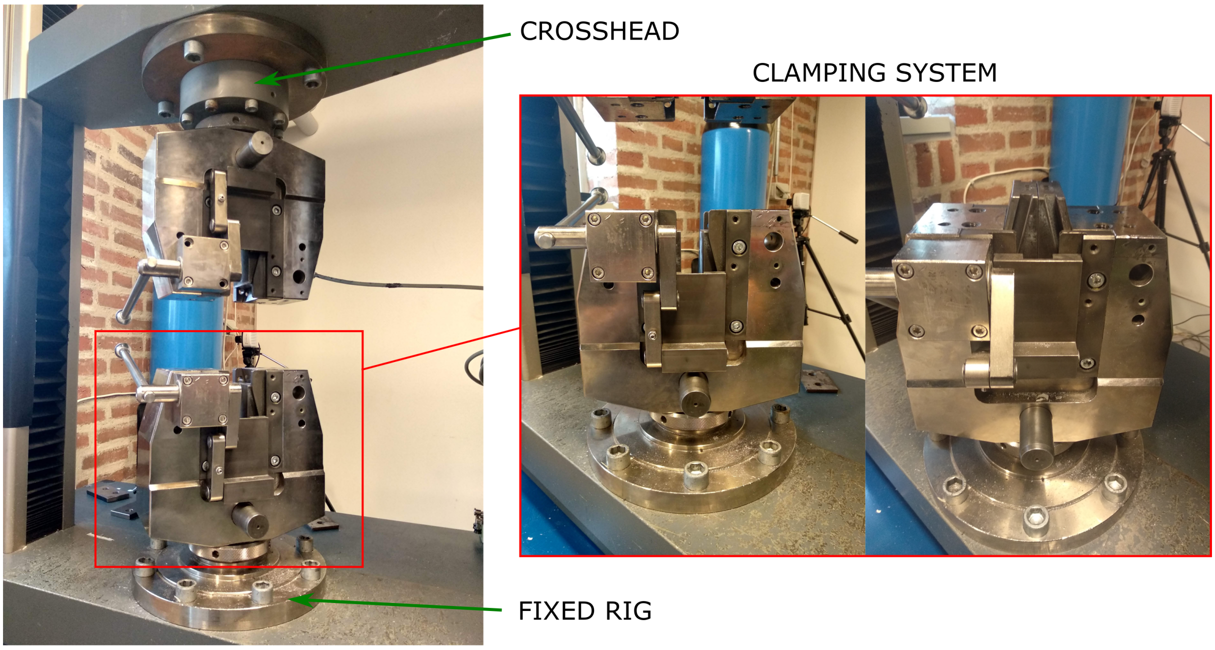
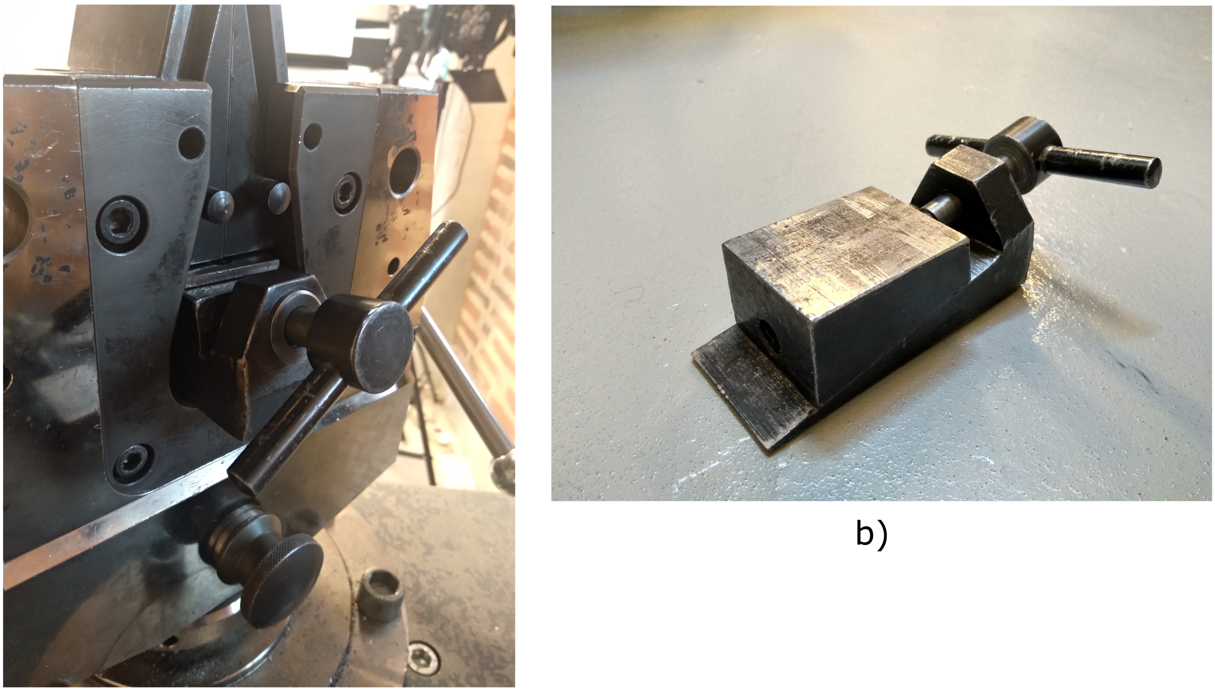


Figure 7.3: At the left side, the crosshead and the rig of the Zwick. At the right side, a detailed view of the clamping system.

It can be the case where the clamp does not have enough grip onto the specimen attached. In this case, a pair of wedges are available. The function of the wedges (see Fig 7.4) is to generate a lever effect to the clamping system. Thus, an undesired movement of the specimen during the test is prevented.



a) b)

Figure 7.4: a) Wedge mechanism placed into the clamping system, acting as a lever. b) Wedge mechanism.

The Zwick is connected to a computer, which has installed the software *TestXpert II*. This program is used to control the machine i.e. the displacement of the crosshead. Different configurations are available depending on the test which is to be performed. The configurations are found in the main directory of the program as *.zp*2 files. In this case, the *tensile test* option is chosen. Although the tensile test is not done as such, since a delamination experiment is performed, the same configuration as for a tensile test is used. Prior to the experiment, the settings of the test must be defined. The options which are available and relevant for the current experiment are stated in the following.

- Speed of the moving crosshead until up to the end of the test.
- Stopping criteria.

When the settings are defined, the test is ready to be commenced. The crosshead is manually set to the start position, where the specimen is attached. For the delamination of the DCB, the crosshead starts to displace upwards at a constant velocity, which has been specified by the user. It can be the case where for convenience, the experiment is stopped prematurely. Otherwise, it continues until the stopping criteria are met.

Once the test is finished, a *.xls* file which contains the data gathered during the delamination process is created. The information that is present in the file has been specified by the user prior to the experiment. The same main *.sp*2 file can be used to run tests for different specimens, thus having various configurations and its respective results at the same place. This interface allows the user to run different tests without having to prepare a new one every time.

7.6 Delamination Experiment

The test specimens, which are already painted and with the hinges screwed, are attached to the headers through the free part of the hinge, see Figure 7.5. The fact of the hinges being thin, makes the grip to have some undesired movement when the specimen is clamped. Therefore, the wedges are added below the clamping system (see Figure 7.4), and this movement is avoided. As it can be seen in the Figure 7.3, the headers are already rotated. With this setup, the pictures for the DIC analyses can be taken properly.

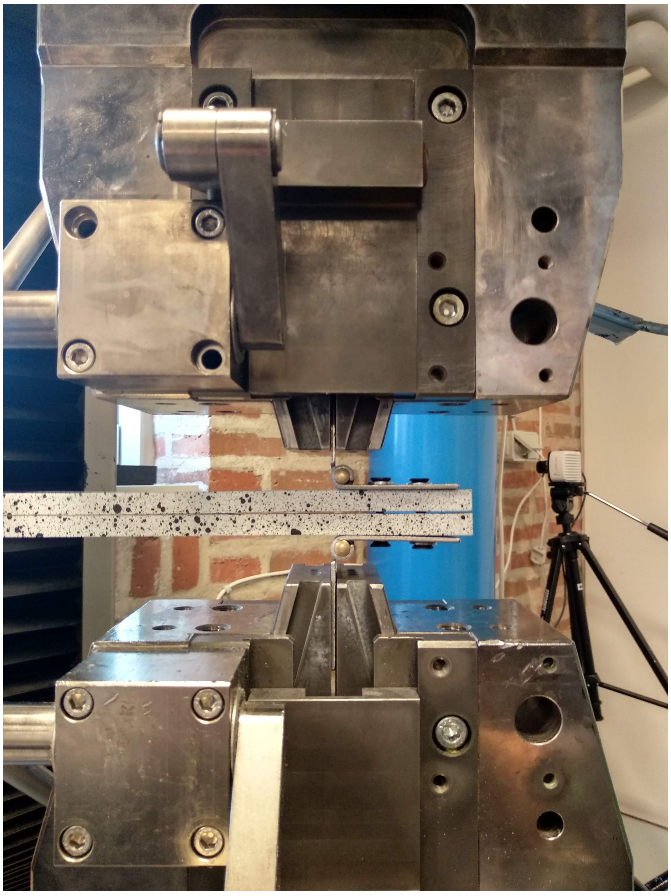


Figure 7.5: DCB specimen attached to the clamping system, before the delamination.

Once the sample is correctly placed, the DIC setup is prepared. The images of the specimen are taken to the lateral face, to capture the deformation of the centerline of the two beams of the DCB while the delamination process is taking place. The camera is put in place, at a distance enough from the specimen so that the picture range can include the fully developed FPZ and some of its development along the interface.

The camera support is situated perpendicular to the specimens lateral face (horizontally) and the vertical angle of the camera is measured with an inclinometer and set to zero. After this, the white light sources are placed behind the camera and at a higher position than the camera, as it can be seen in Figure 7.1. It must be assured that the illumination of the surface assures a good contrast between white and black throughout all the duration of the test [Jones and Iadicola, 2018]. With the lighting, the shutter time of the camera is set so that there are no overexposed/underexposed areas in the image taken, that introduce noise to the DIC results [Jones and Iadicola, 2018]. Then, the number of images to be taken is selected and the rate images per second is set. After this, the image acquisition process starts while the crack is first seen at the edge of the camera view. Note that the crack growth rate has been roughly estimated in a previous non-recorded test to see how much time the crack takes to cross the camera view.

Figure 7.6: Image showing the painted specimen under delamination.

The parameters which are introduced in both the Zwick software prior the experiments are stated in Table 7.2. As it has been stated at the beginning of this chapter, the value which is critical for the performance of the experiment is the displacement rate i.e. how fast are the arms of the DCB specimen separating. From the work presented last semester in Viejo et al. [2019], it was shown that a too fast vertical displacement could provoke the interface to delaminate violently. As a result of the fast delamination, most of the pictures taken by the DIC had to be discarded. The desire of having a stable crack growth, implies a low value of opening rate. The opening speed is set to be 15 mm/min. This value is chosen from a previous test, where it was observed that the delamination process is smooth. This also implies that the pictures of the DIC will be less probable to be moved.

As it can be seen in the table above, the stopping criterion is established as the upper force limit. This is the maximum force allowed during the test. Since it is desired to stop the test manually, the value of the force is set deliberately high. This done in order to prevent the experiment to be stopped by the software, as this force value is never reached.

Due to the weight of the specimen and the axial adjustment of the hinges, when the DCB is unloaded, the specimen lays out of the plane where the delamination experiment occurs, and where the DIC pictures are taken. Therefore, it is decided to apply the force needed to reallocate the specimen to the correct plane, before any picture is taken. In the case of not doing this, the reference picture (which is used to correlate the displacement field), would be taken in a distinct plane from the one where the vertical displacements are happening, and the whole correlation analysis would be wrong.

Regarding the DIC parameters introduced in the software ARAMIS, it is decided to take 120 pictures in 120 seconds. The user must only be sure that the pictures taken have the crack fully developed and that the crack does not go near the left end of the camera field of vision. If this happens, tractions behind the crack tip might be located "out of the picture" and are not going to be captured in the displacement field obtained, making the algorithm to compute wrong results. This is especially important since the fracture process zone size is unknown. The shutter time is set to 10 ms. However, it is strongly dependent on the lightning used, so it must be adjusted depending on each experiment.

	Specimen 1	Specimen 2
Test speed (mm/min)	15	15
Upper force limit (N)	5000	5000

Table 7.2: Parameters of the Zwick for the performance of the tests.

Therefore, the test can be initiated. When the start button is pressed in *TextXpert II* the crosshead starts to move. Then, the camera is shot and it starts taking pictures of the delamination process. The test lasts until the fracture process zone moves out of the scope of the lens. When this point is reached, the Zwick is stopped manually and the experiment is finished. Then, the crosshead is displaced to the original position, and the delaminated DCB can be removed. This procedure is repeated successively for all the specimens available.

8 | Results and Discussion

In this chapter, the experimental data obtained during the experiment explained in Chapter 7 are processed and discussed. A summary of the postprocessing of the images to obtain the displacement field used as input is given. Results of the cohesive law determined using the inverse parameter identification procedure are shown and analysed, both for the linear formulation and the quadratic formulation. Finally, the results for the cohesive law confidence intervals are obtained with the aid of a MC simulation.

8.1 DIC-Results

The images taken during the delamination experiment are processed to obtain the displacement values in the same manner as in the previous semester project Viejo et al. [2019]. The program used is GOM Correlate on its free version. The images are analyzed after the correction of the distortion of the lens, see Section 7.4.

After importing the images to the program, the relation unit length/pixel is defined using a picture taken with a ruler on the plane of the lateral face of the specimen, see Figure 8.1

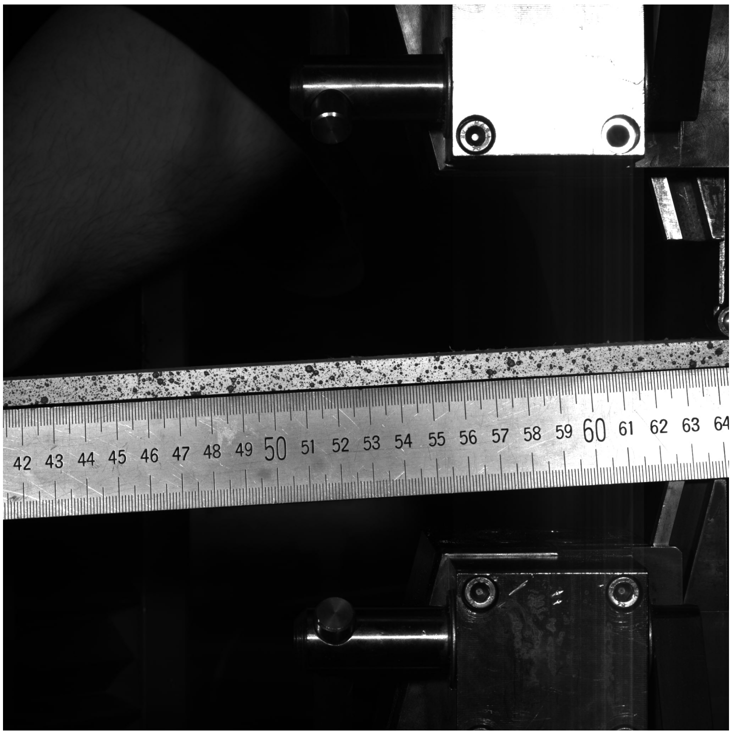


Figure 8.1: Image used to set the relation unit length/pixel.

A region of interest (ROI) is defined at the specimen lateral face. The displacements in the vertical direction are calculated on the whole ROI. The software recalculates the position of the ROI, being able to adapt it to the crack growth.

A curve is defined on the centerline of each beam, as it can be seen in Figure 8.2.

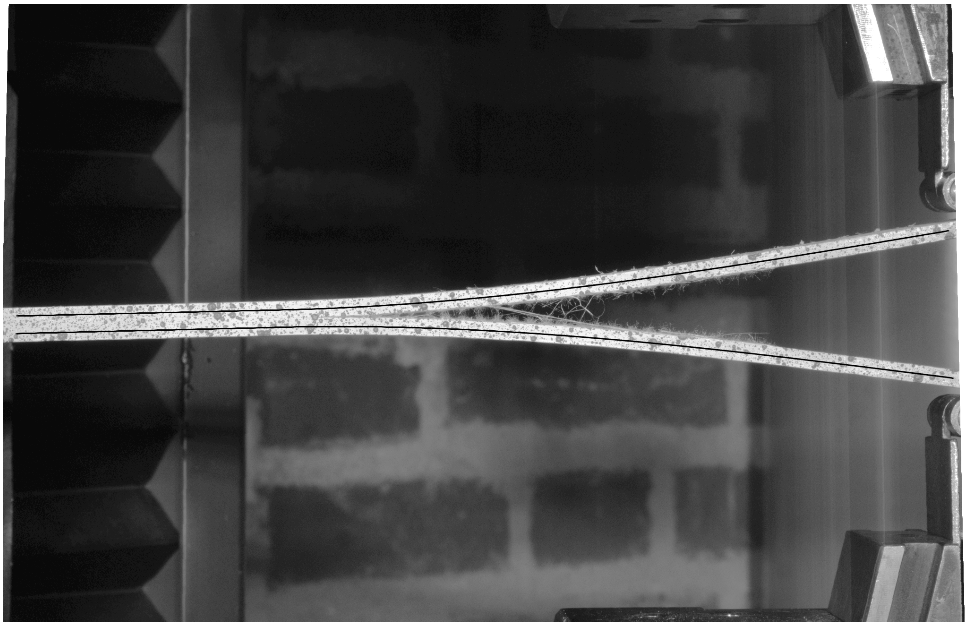


Figure 8.2: Curves used to export the vertical displacement of the centerline in GOM Correlate.

The vertical displacement is calculated at each of the lines. The difference in displacement in between the upper and the lower beams is calculated for each point obtaining the separation curve of the beams for each picture. An example of the obtained curve is shown in Figure 8.3.

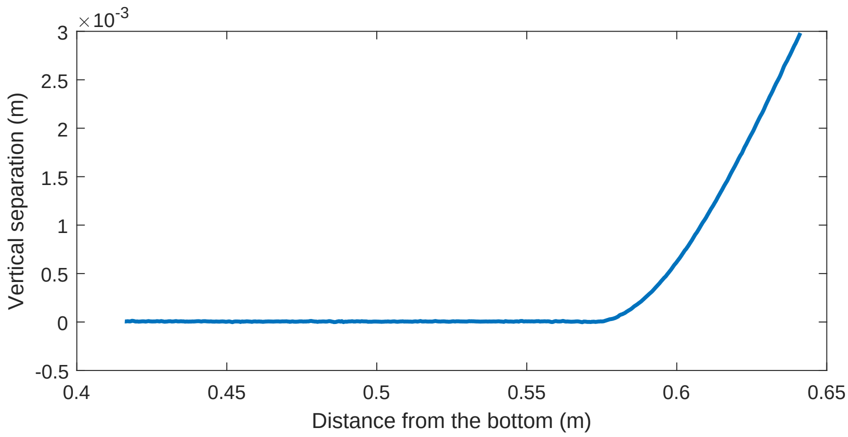


Figure 8.3: Vertical separation in between the upper and lower beams.

In Chapter 2, it is explained how the difference in vertical displacement between the delamination interface and the centerline, where the DIC measurements are taken, biases the calculated cohesive law. It is shown how it can be overcome by using the separation values at the interface as separation values of the cohesive law while the displacement of the centerline is used for the inverse parameter identification of the tractions. This could not be done because the resolution of the image does not allow the DIC calculation to get close enough to the interface. Therefore, the displacement at the centerline is used instead. If this is to be performed in a proper manner, a camera with higher resolution should be used or the pictures should be taken closer to the specimen.

The calculation of the vertical displacements is performed with four different settings displayed in Table 8.1. The choice of the *Point distance* is linked to the choice of the *Facet size*. The *Point distance* is always chosen to be the integer immediately above the 50% of the *Facet size*, as this combination provides the best accuracy [GOM mbH, 2015]. The calculation only succeeded for the *Facet size* values in the table, failing for other values because of a deficient correlation.

Facet size (pixels)	9	11	13	15
Point distance (pixels)	5	6	7	8
Subpixel interpolation	Bicubic	Bicubic	Bicubic	Bicubic
Max. sampling points	Inf	Inf	Inf	Inf
Facet matching	Against previous stage	Against previous stage	Against previous stage	Against previous stage
Interpolation size	0	0	0	0

Table 8.1: Different settings used in GOM Correlate for the DIC analyses.

The choice of the *Facet size* which is to be used for the inverse parameter identification of the interfacial tractions field involves a trade-off between precision and noise. A smaller *Face size* provides a more precise calculation because more points are used for the correlation of the images, and so, a more detailed displacement field is provided. Whereas a bigger *Facet size* results in a more smooth displacement field but less detailed. As a result, it is decided to study the impact of the chosen *Facet size* over the results in the following.

8.1.1 Impact of the *Facet Size* over the Calculated Tractions

The results of the inverse parameter identification of the tractions when the displacements are obtained with the four settings displayed in Table 8.1 are compared in Figure 8.4 and 8.5, corresponding to the interfacial tractions and the cohesive law, respectively.

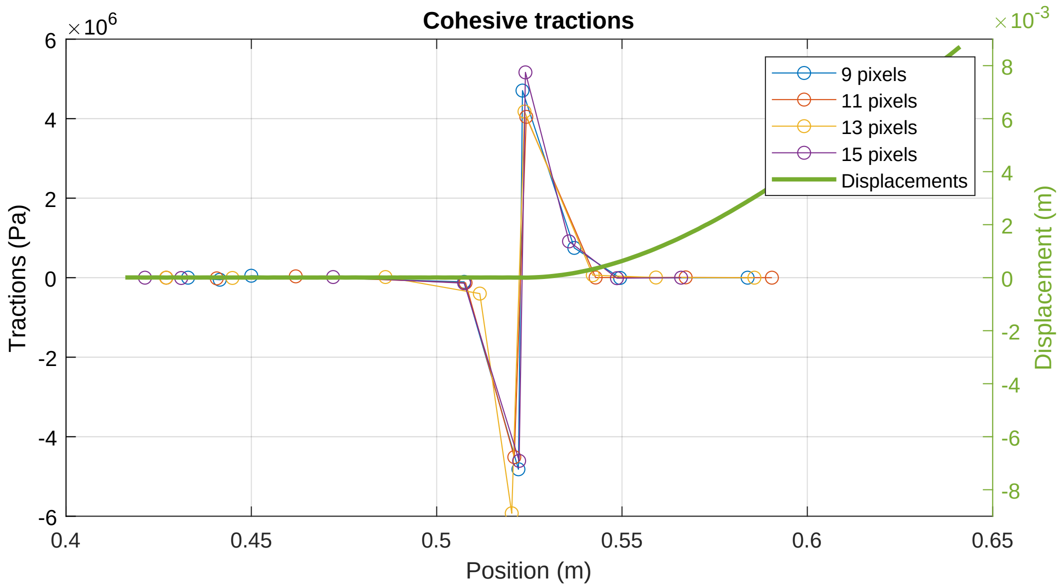


Figure 8.4: Traction field calculated for different *Facet sizes* and the displacement curve corresponding to the picture used. Seven d.o.f. are used for the traction field.

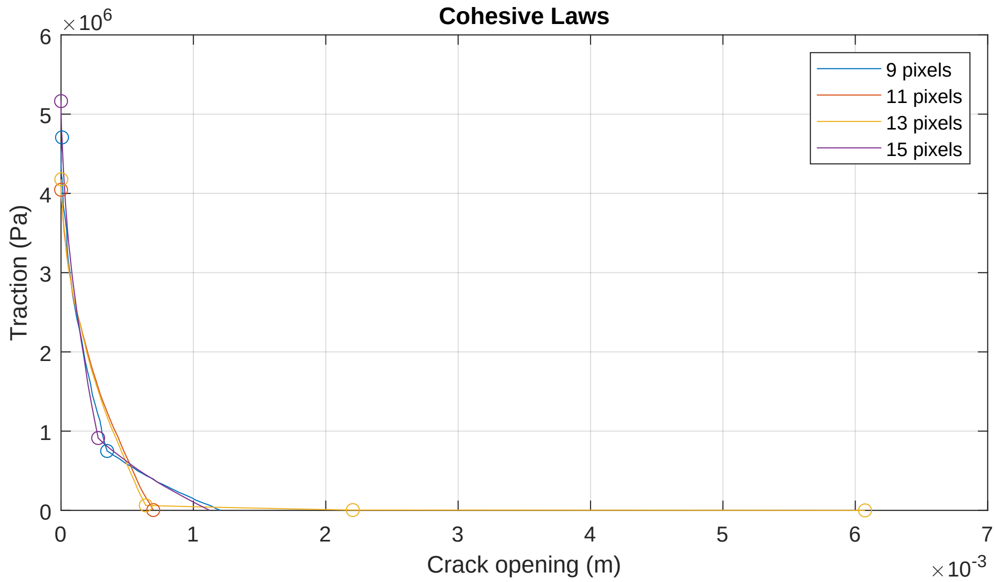


Figure 8.5: Cohesive law calculated for different *Facet sizes* and the displacement curve corresponding to the picture used. The displacement curve corresponds to a *Facet size* of 15 pixels. Seven d.o.f. are used for the traction field.

Any deviation in between different *Facet sizes* is attributed to the deviations in displacements caused by the inherent noise of the DIC measurements and the difference in the location of the data points. No pattern is seen in the variation of the results for other number of d.o.f. nor using the constraint quadratic loading functions. Therefore, it is assumed that the subset size does not have an impact on the results for the range used.

8.2 Triangular Loading Functions with Force Calculation

In this section the results of the inverse parameter identification algorithm with triangular load functions, a more developed version of the algorithm created in the previous semester in Viejo et al. [2019] (explaiend in Section 6.1) , are shown for the DIC pictures. The results are shown for 7 traction d.o.f. for simplicity, but a similar behaviour is observed with all the number of traction d.o.f. used until reaching the point where the inverse parameter routine becomes unstable. In the same way, aiming for a simple and clear presentation of the results obtained, out of the 51 DIC pictures that have been post-processed, the tractions (see Figure 8.6) and cohesive laws (see Figure 8.7) presented correspond to pictures 15, 25, 35 and 45 respectively.

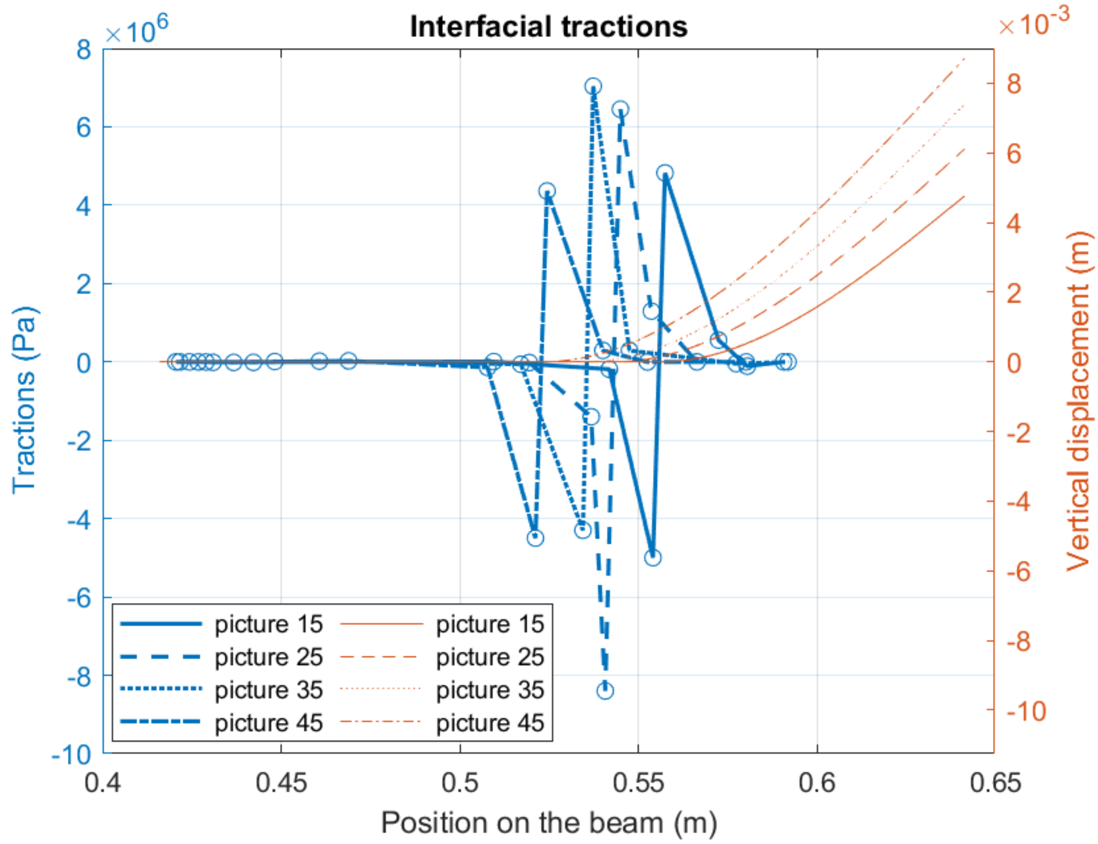


Figure 8.6: Computed interface tractions (in blue) for the displacement fields of the pictures selected (in orange).

It is observed in Figure 8.6 how the traction shapes predicted by the algorithm are similar. The cohesive laws in Figure 8.7 also show a similarity in shape but both the onset traction and onset displacement predicted have a relatively big discrepancy between displacement fields for different DIC pictures.

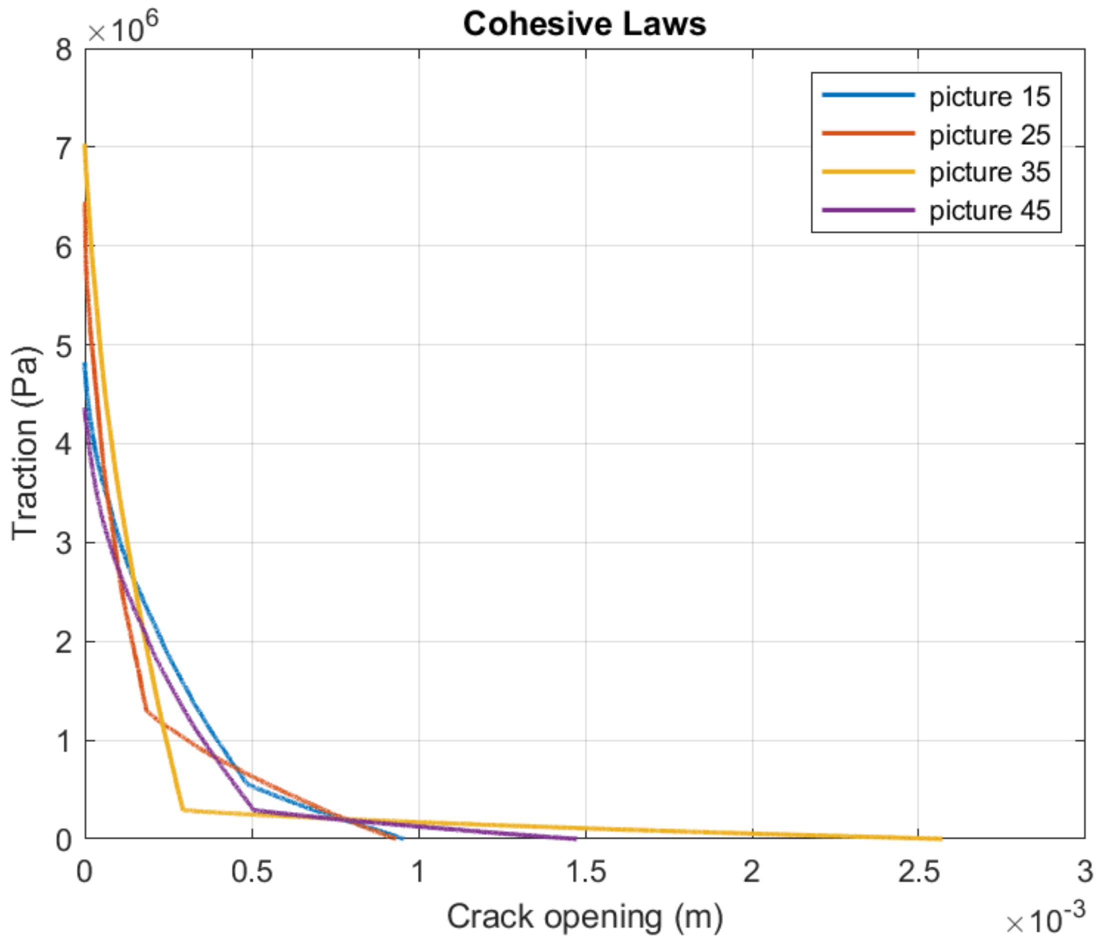


Figure 8.7: Cohesive laws obtained for the tractions calculated in Figure 8.6

The values of the onset traction predicted for all the DIC pictures are shown in Figure 8.8 and show the main limitation of this formulation using triangular linear load functions, which is the accuracy on the prediction of the onset traction value (maximum value is 10 MPa and minimum is 3.5 MPa). Moreover, the energy release rate, calculated as the area below the cohesive law, is also shown in Figure 8.8, which varies from 950 J/m^2 to 1180 J/m^2.

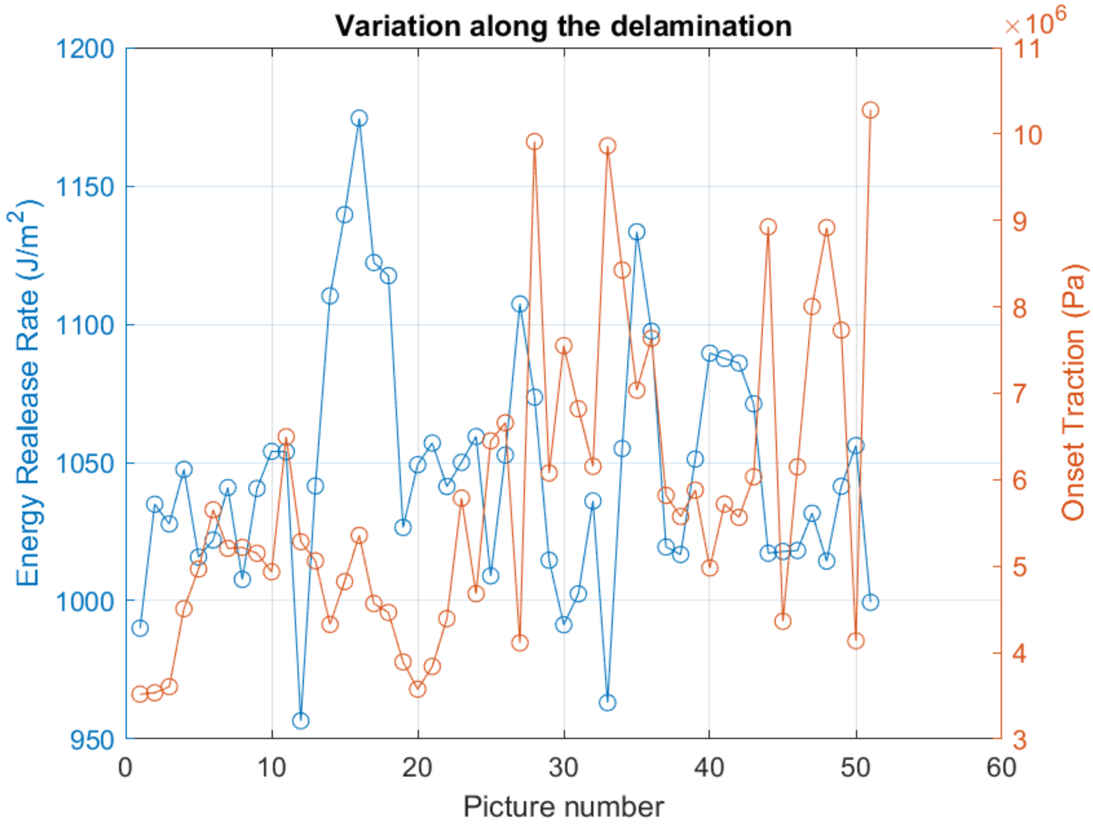


Figure 8.8: Energy release rate (blue) and onset traction (orange) for each picture taken.

Even though the force exerted by the tensile test Zwick machine is not used for any calculation, the data of the experiment has been extracted to have some bounds for the force predicted (see Figure 8.9). This allows to have a guess on whether the algorithm is working correctly or not.

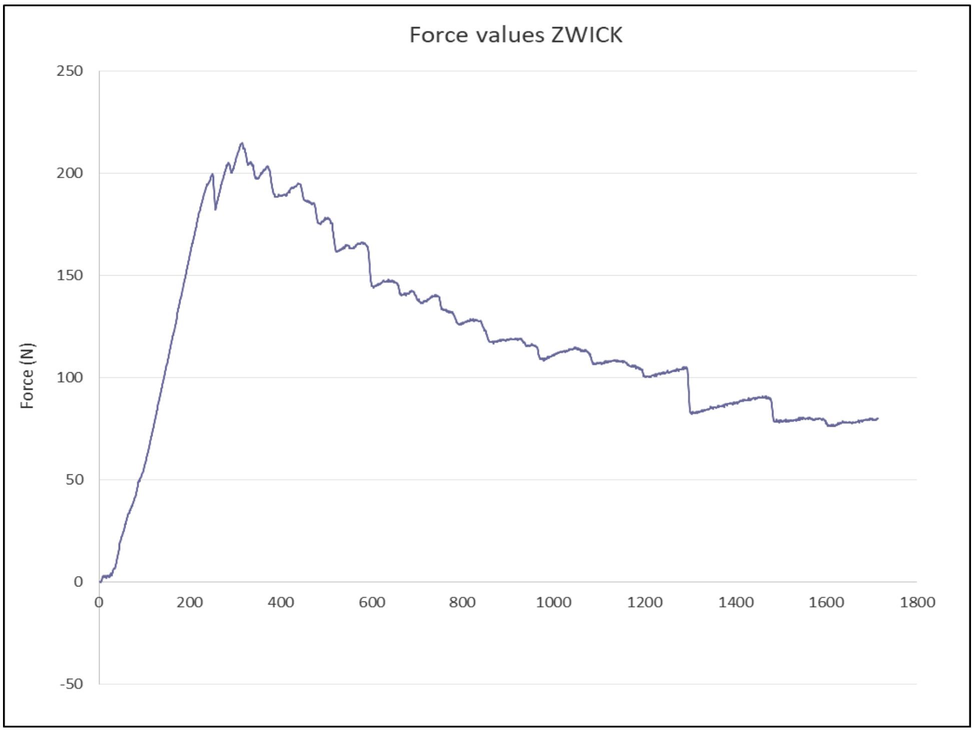


Figure 8.9: Force data obtained from the Zwick machine. Note that the x axis just represents the number of measurements of the force done and not the time, which was not measured.

It is seen in Figure 8.10 that the force predicted by the algorithm is decreasing as the experiment is being performed, just as Figure 8.9 shows. Moreover, the values for the force predicted in Figure 8.10 are between 230N and 130N approximately. These values are close to the values seen on Figure 8.9 (where the maximum force is 220 N), which suggests that the forces that are predicted by the inverse parameter algorithm are a good representation of the reality.

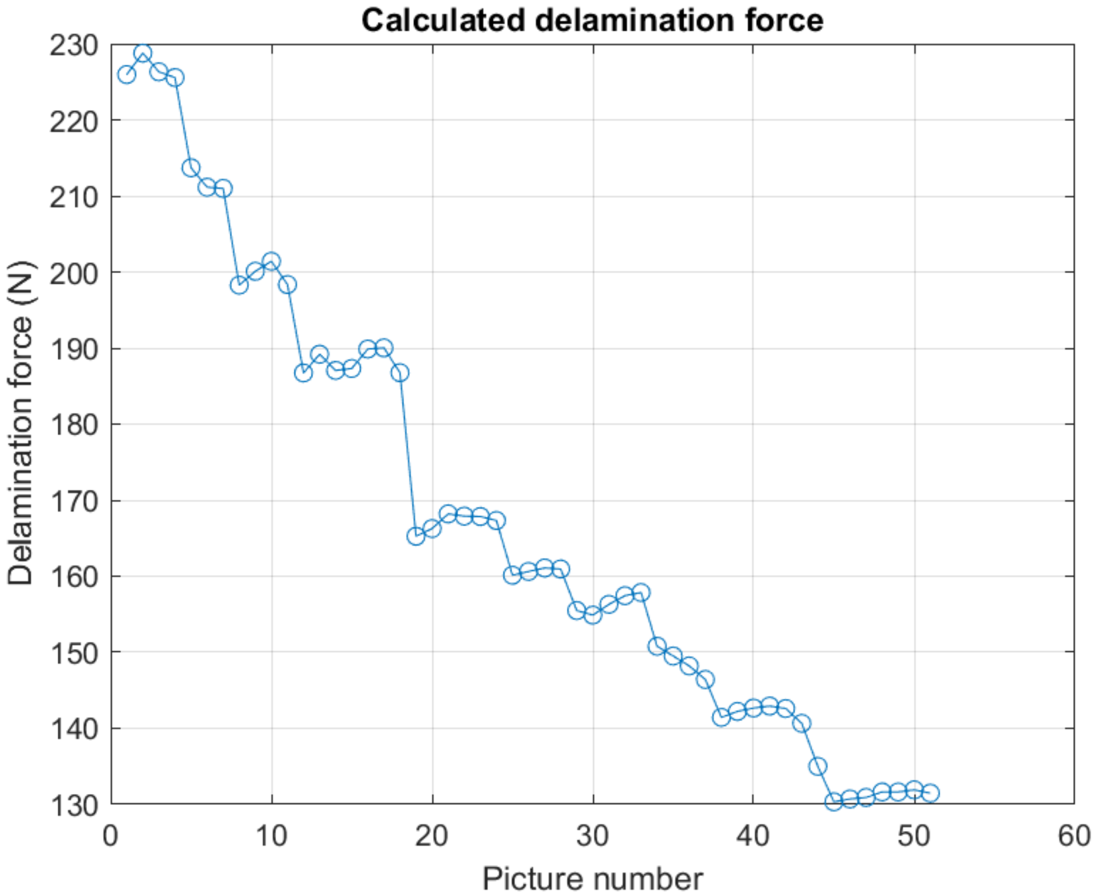


Figure 8.10: Force predicted by the inverse parameter algorithm for each picture taken.

8.3 Quadratic and Triangular Loading Functions

The current section presents the results obtained from the inverse parameter identification routine, with the combination of linear and quadratic loading functions and the linear constraints, introduced in Chapter 6. Although a similar behaviour of the results is observed from 8 to 16 traction d.o.f., the following images display the results obtained for 8 traction d.o.f. Figures 8.11 and 8.12, show the outcome from four different pictures, taken out from the 51 DIC images which have been postprocessed. Figure 8.11 displays the interfacial tractions, while Figure 8.12 shows the obtained cohesive laws, for four different pictures (15, 25, 35, and 45) of the delamination process.

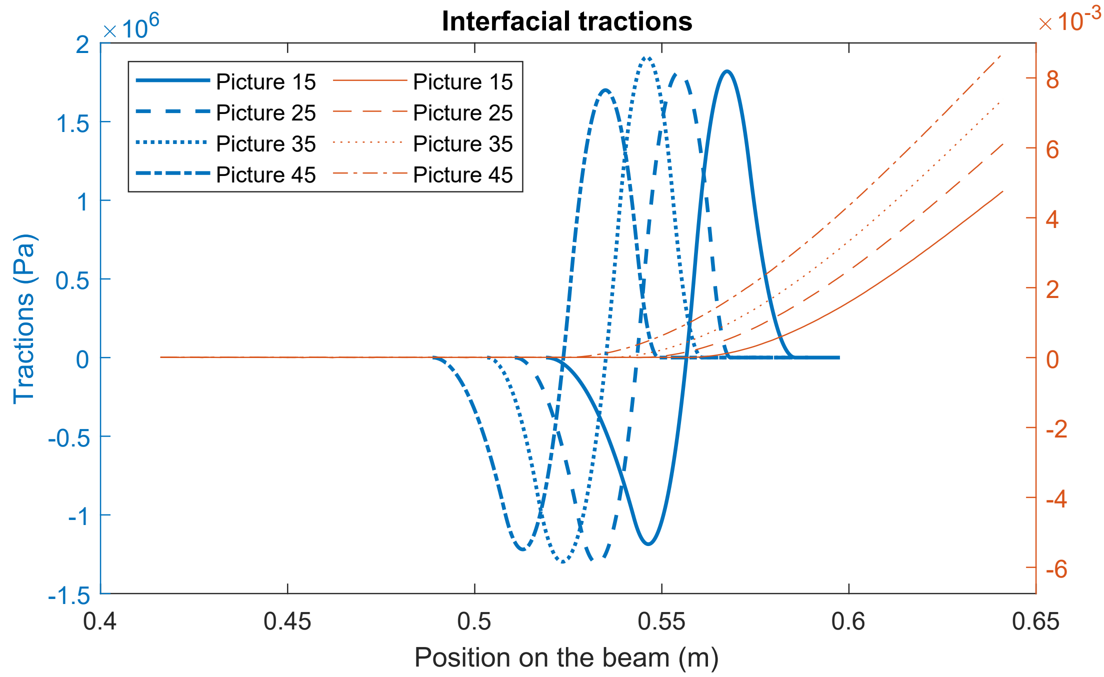


Figure 8.11: Calculated interfacial tractions (in blue) for the displacement fields of the pictures selected (in red). Each line style corresponds to a picture.

It is observed that the shapes of the different interfacial tractions in Figure 8.11 are very similar; the maximum and minimum values for the traction are closer between them than in Figure 8.6. As for in Figure 8.12, the four cohesive laws have high resemblance both in shape and onset traction. The values of the onset traction are remarkably lower than the ones observed in Figure 8.7.

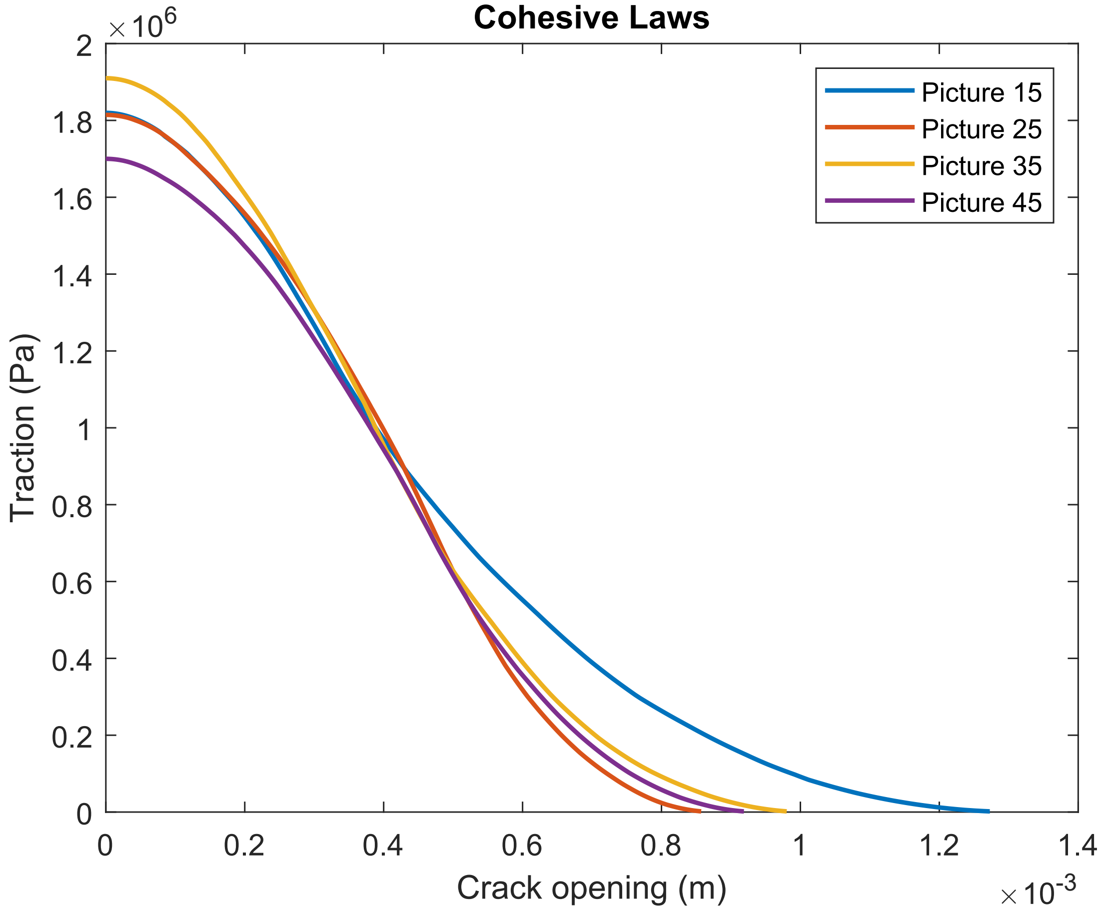


Figure 8.12: Calculated cohesive laws from the interfacial tractions displayed in Figure 8.11.

Figure 8.13 shows the evolution of the limits of the fracture process zone during the delamination process, once the crack is fully developed. Red dots mark the position of the point where full damage of the interface is achieved. Blue dots show the position of the maximum traction, and the yellow dots mark the starting position of the zero compressive interfacial tractions. A slight increment in the limits of the FPZ is noticed as the delamination goes on.

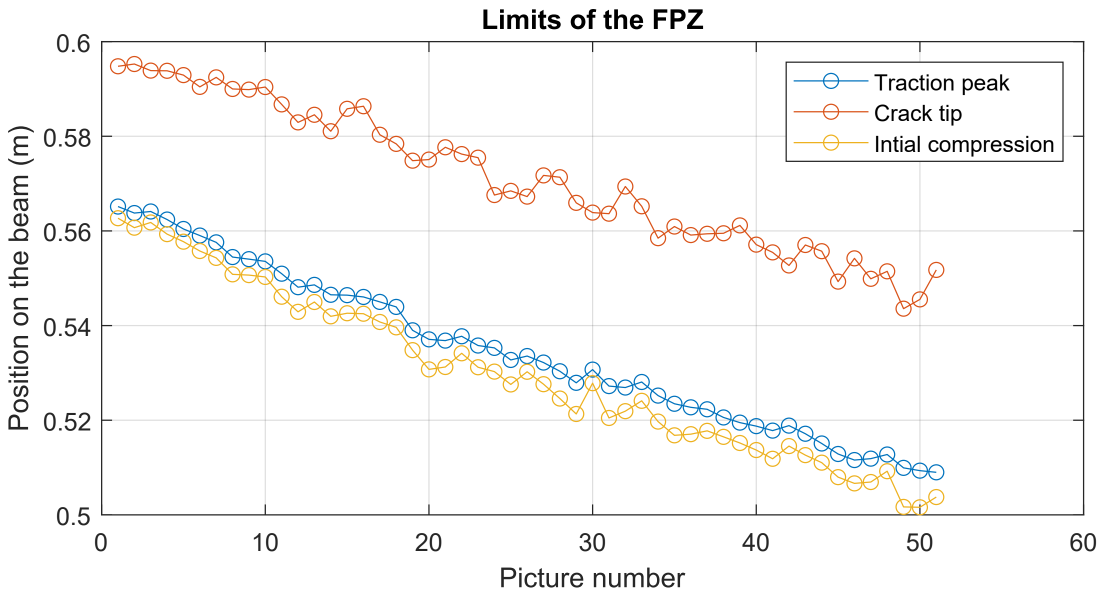


Figure 8.13: Translation of the FPZ during the delamination process.

Displayed below, Figure 8.14 shows the values of the onset traction predicted for all the 51 DIC images mentioned above, together with the variation of the energy release rate. Compared to Figure 8.8, the variation of the values of the onset traction is more stable, while for the energy release rate, the variation is the same, but with lower values. Onset traction values are much less dispersed, as they vary in between 1.5 and 3 MPa. This can be related to the better adaptability of the new loading functions, and the constraints limiting the freedom of the tractions shape. On the other hand, the energy release rate also fluctuates around a lower value, as a consequence of the onset traction being lowered.

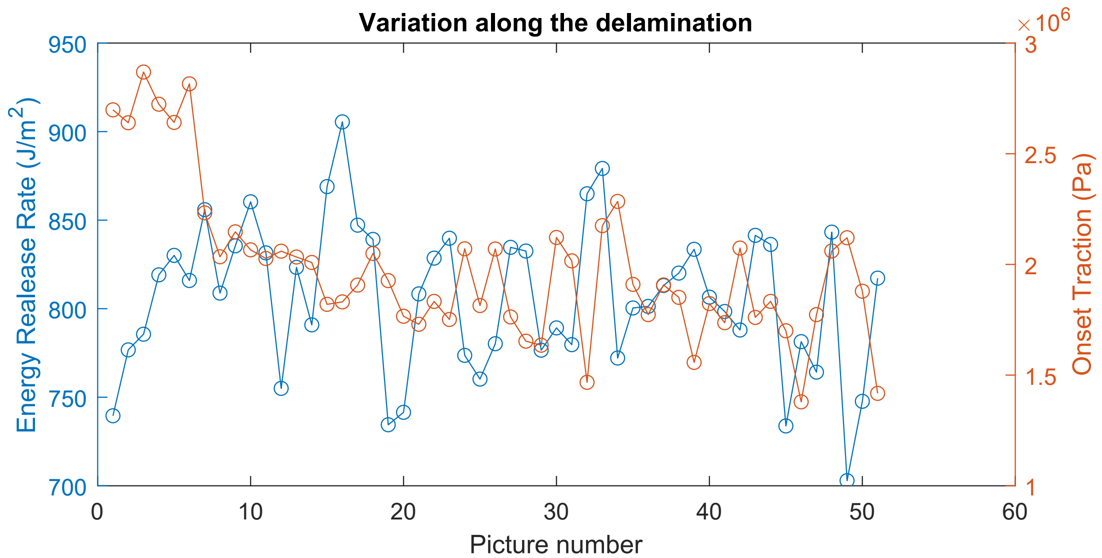


Figure 8.14: Force predicted for each picture taken.

The evolution of the force calculated with the new inverse parameter identification tool is almost the same as the one captured in Figure 8.10. Therefore is not shown here. The maximum and minimum values are similar to the ones extracted from the Zwick machine.

8.4 Monte Carlo Results

The MC algorithm displayed in Chapter 4 has been used to evaluate the sensitivity of the outcome of an inverse parameter identification routine when the input is subjected to uncertainty. The result is compared to the cohesive law introduced to the FE-model to assess the performance of each routine. Applying it to the experimental results, the precision of the obtained results cab be assessed. The algorithm is the same as the one in Chapter 4, the only change is the input. Now the displacement data is the one obtained by means of DIC, for the experimental procedure see Chapter 7.

Two different MC simulations are performed:

- **Assessment of the method:** a simulation in which the input displacement data does not change in between samples. Only one picture of the DIC is used in the whole simulation. The precision of the method is assessed in the same conditions as in Chapter 4 but against real experimental data.
- **Material characterization:** a simulation in which the number of DIC picture is included as a random variable in the MC algorithm. The vertical displacement data changes in between samples. The resulting variability of the cohesive law includes, not only the variability due to the uncertainty of the input parameters and the instability of the inverse parameter identification routine, but also the variability of other parameters not included in the random inputs of the MC algorithm, e.g. the heterogeneity of the material.

8.4.1 Assessment of the Method

The displacement data of the picture 25 of the delamination experiment of Chapter 7 is used with the two selected inverse parameter identification models. The random variables included in the MC simulation are displayed in Table 8.2

Parameter	Distribution	Mean	Standard deviation
Flexural rigidity EI	Normal	78.15 Nm^2	3.44 Nm^2
Shear rigidity kAG	Normal	462000 N	46200 N
Width w	Normal	0.0287 m	5.9×10^{-5} m
Force application point L	Normal	0.64 m	0.0019 m

Table 8.2: Random variables used in the MC simulation when only one picture is considered for the displacements.

The resulting cohesive law for the triangular loading model, which also calculates the force is displayed in Figure 8.15. The obtained cohesive laws have been separated in two plots: one showing the steepest part of the cohesive law and the other the part of large separation values.

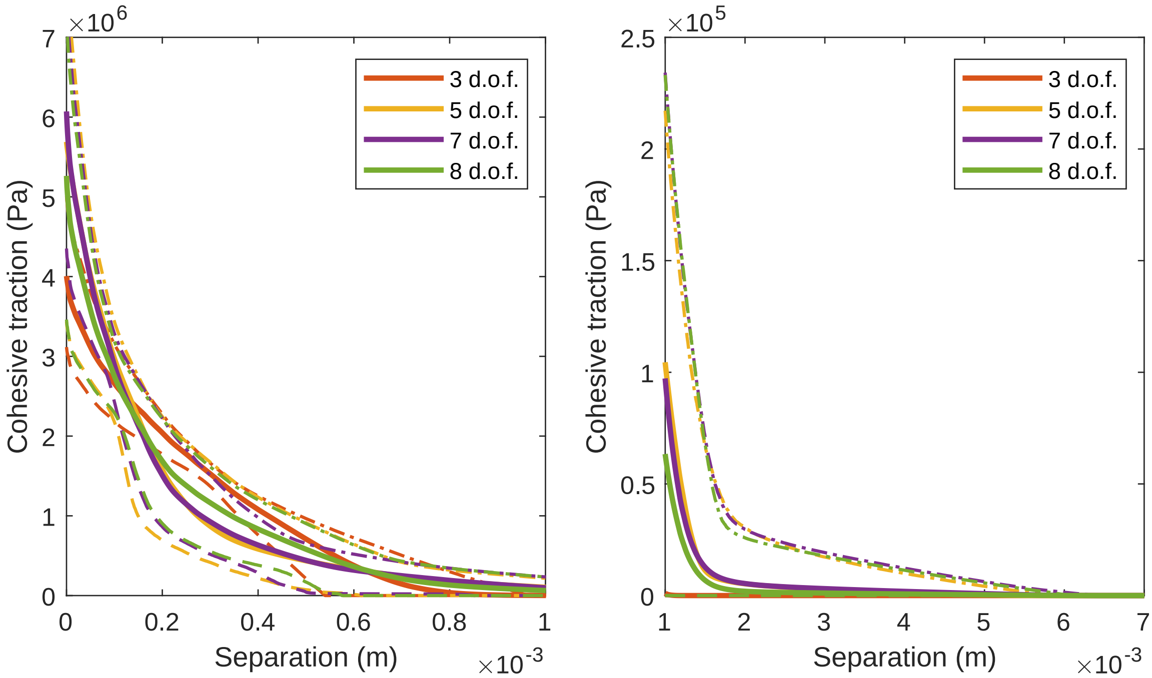


Figure 8.15: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{LB,j}^M$ (dashed-dot line), obtained from the MC algorithm using 3, 5, 7 and 8 traction d.o.f. with the triangular loading functions. The legend shows the traction d.o.f. used to calculated each of the curves.

The same behaviour as in Chapter 4 is seen, where the mean cohesive law $[\overline{\sigma}_j^M, \delta_j]$ tends to a certain curve as the number of traction d.o.f. increases, and remains stable after a certain amount. Adding more traction d.o.f. when the mean cohesive law $[\overline{\sigma}_j^M, \delta_j]$ does not change, only widens the confidence interval $[\sigma_{LB,j}^M, \sigma_{UB,j}^M]$. In Figure 8.15, the mean cohesive law $[\overline{\sigma}_j^M, \delta_j]$ barely changes in between 5, 7 and 8 traction d.o.f. However, the confidence interval $[\sigma_{LB,j}^M, \sigma_{UB,j}^M]$ is narrower for the results with 7 traction d.o.f., so it is considered the optimum number of traction d.o.f. for this case.

The same result using the quadratic loading model is shown in Figure 8.16.

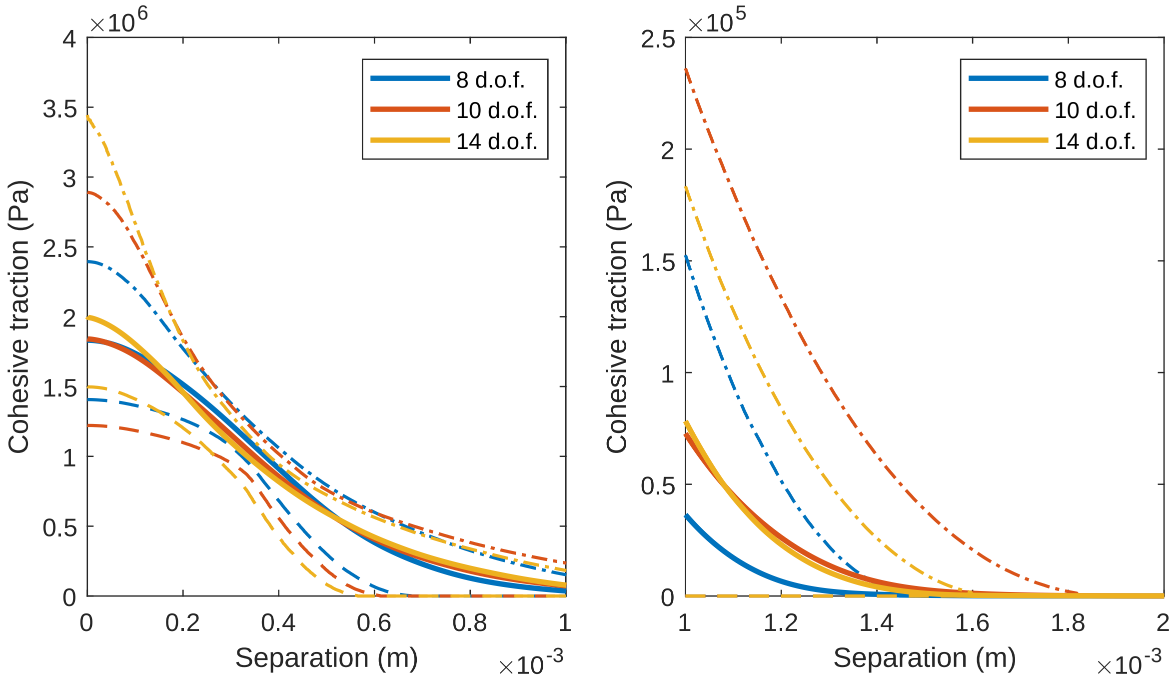


Figure 8.16: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{LB,j}^M$ (dashed-dot line), obtained from the MC algorithm using 8, 10 and 14 traction d.o.f. with the quadratic loading functions. The legend shows the traction d.o.f. used to calculated each of the curves.

The minimum number of traction d.o.f. the quadratic model can use is 8, because for a lower number of traction d.o.f. the number of constraints is bigger than the number of traction d.o.f. The narrowest confidence interval $[\sigma_{LB,j}^M, \sigma_{UB,j}^M]$ is the curve corresponding to 8 d.o.f. Therefore, the optimum number of d.o.f. is considered to be 8.

The results for the triangular loading functions model for 7 traction d.o.f. and the quadratic loading functions model for 8 d.o.f. are plotted in Figure 8.17.

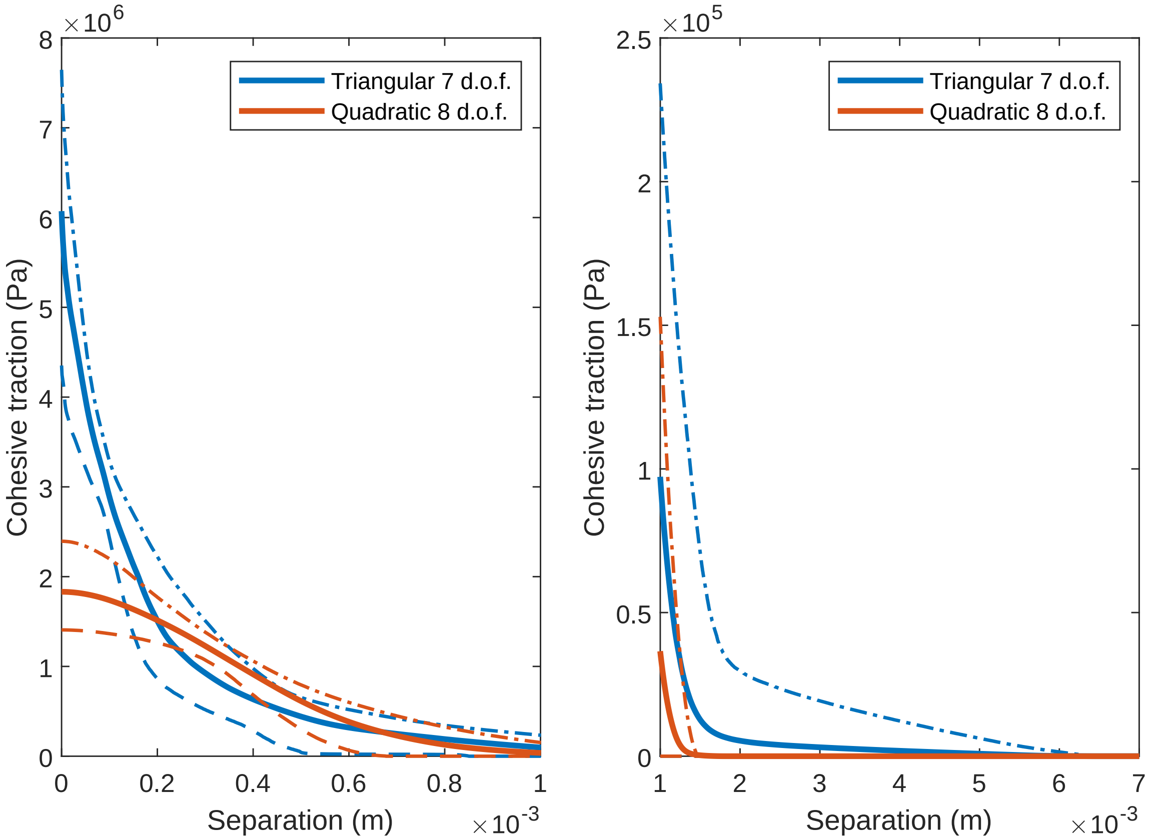


Figure 8.17: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{LB,j}^M$ (dashed-dot line), obtained from the MC algorithm using 7 and 8 traction d.o.f. with the triangular loading functions and the quadratic loading functions, respectively. The legend shows the traction d.o.f. and the loading functions used to calculated each of the curves.

Shown in Chapter 6, the quadratic loading functions model predicts a lower onset traction and a smaller final separation because its constraint of C^1-continuity does not allow for abrupt changes that might be present in reality. On the other hand, the triangular loading functions model predicts a higher onset traction and a larger final separation. This is caused by a low number of traction d.o.f. used to represent piece-wise linearly a complex curve with several changes in slope.

Unexpectedly, the confidence intervals for both solutions are of the same magnitude, even though the quadratic loading function model showed a clearly superior performance in Section 6.4. This result might be related to the shape of the material cohesive law which could be more easily represented by the triangular loading functions. Moreover, it could also be related to the convexity of the nonlinear optimization, the optimization of the load positions, design space of the quadratic constrained loading functions. It was shown in Section 6.3.1 that the result of this algorithm is much more dependent on the initial guess of the positions than the triangular loading model. This topic needs from further investigation.

8.4.2 Material Characterization

To include other parameters which may vary in between pictures, e.g. DIC camera noise, and give an overall assessment of the material cohesive law, it is decided to include all the pictures of the DIC in the simulation. This is done by including the number of picture taken throughout the delamination of a whole specimen as a random variable in the MC simulation. The random variables of the new analysis are displayed in Table 8.3.

Parameter	Distribution	Mean	Standard deviation	Interval
Flexural rigidity EI	Normal	78.15 Nm2	3.44 Nm2	-
Shear rigidity kAG	Normal	462000 N	46200 N	-
Width w	Normal	0.0287 m	5.9e-5 m	-
Force application point L	Normal	0.64 m	0.0019 m	-
Picture number	Discrete uniform	-	-	[1,51]

Table 8.3: Random variables used in the MC simulation when all the pictures are included in the simulation.

The results are shown in Figure 8.18 for the triangular load functions and in Figure 8.19 for the quadratic constrained loading functions.

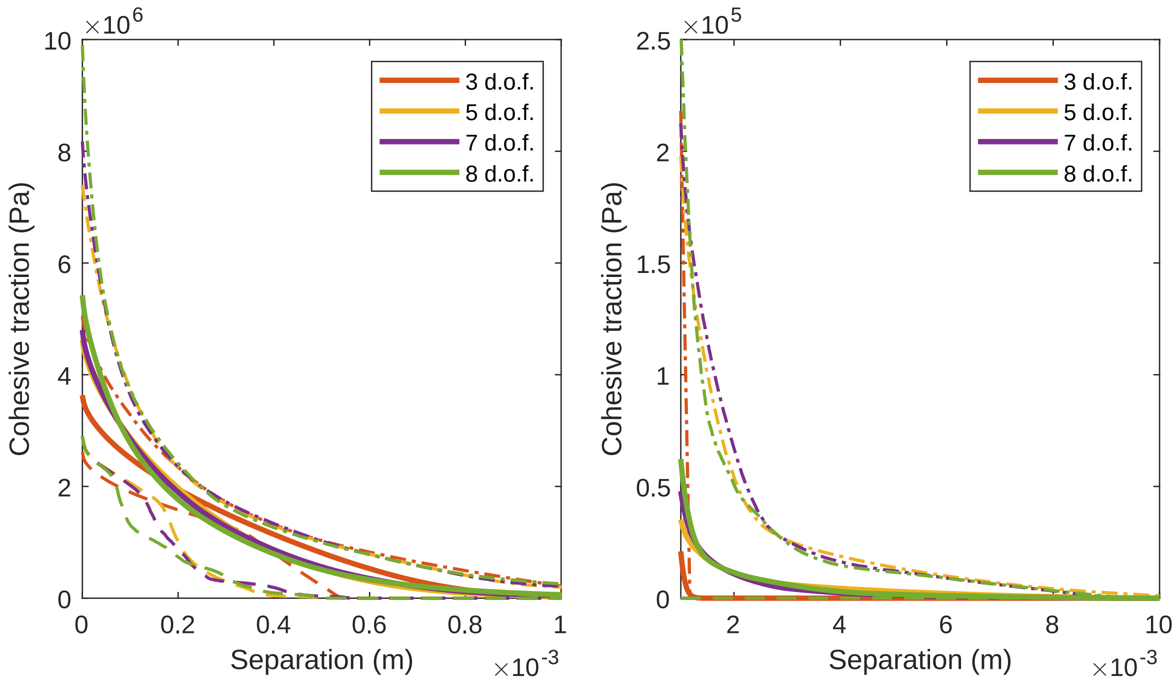


Figure 8.18: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{LB,j}^M$ (dashed-dot line), obtained from the MC algorithm using 3, 5, 7 and 8 traction d.o.f. with the triangular loading functions. The legend shows the traction d.o.f. used to calculated each of the curves.

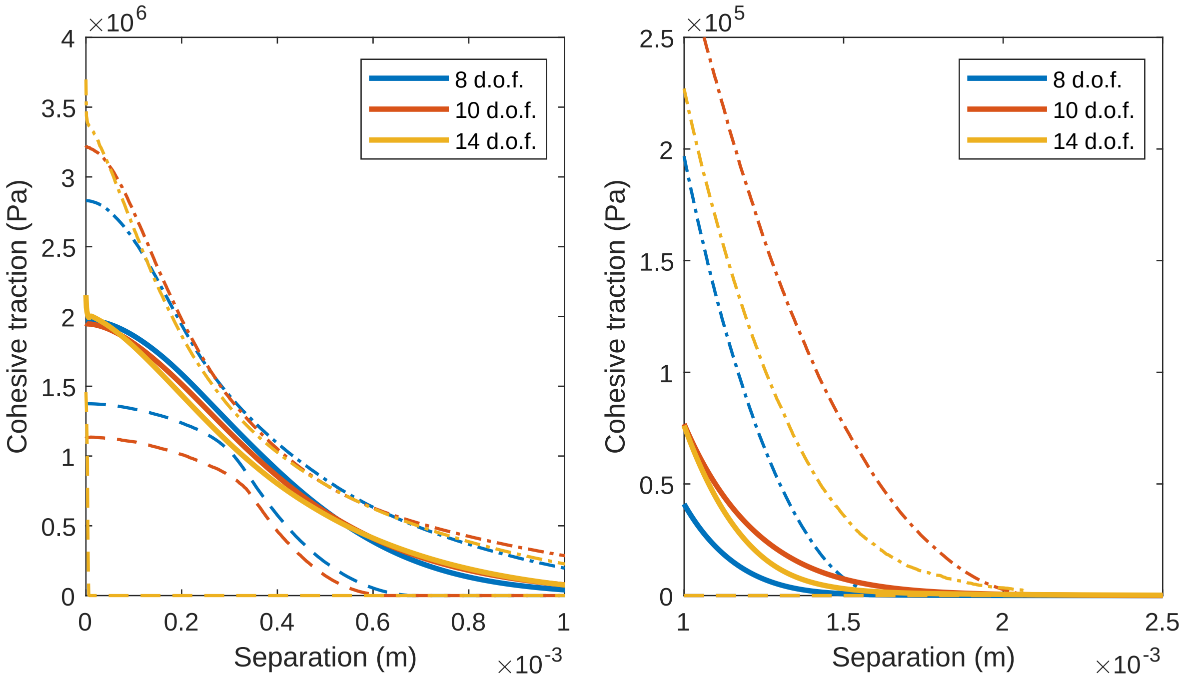


Figure 8.19: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma_{LB,j}^M$ (dashed line) and the upper bound $\sigma_{LB,j}^M$ (dashed-dot line), obtained from the MC algorithm using 8, 10 and 14 traction d.o.f. with the quadratic loading functions. The legend shows the traction d.o.f. used to calculated each of the curves.

The behaviour observed in the results of a single picture is repeated in these analyses. The only difference is the wider confidence interval, which is assumed to take into account the aforementioned parameters which can vary in between the different pictures taken.

The best results for both loading models are obtained for the same number of traction d.o.f. The results are compared in Figure 8.20.

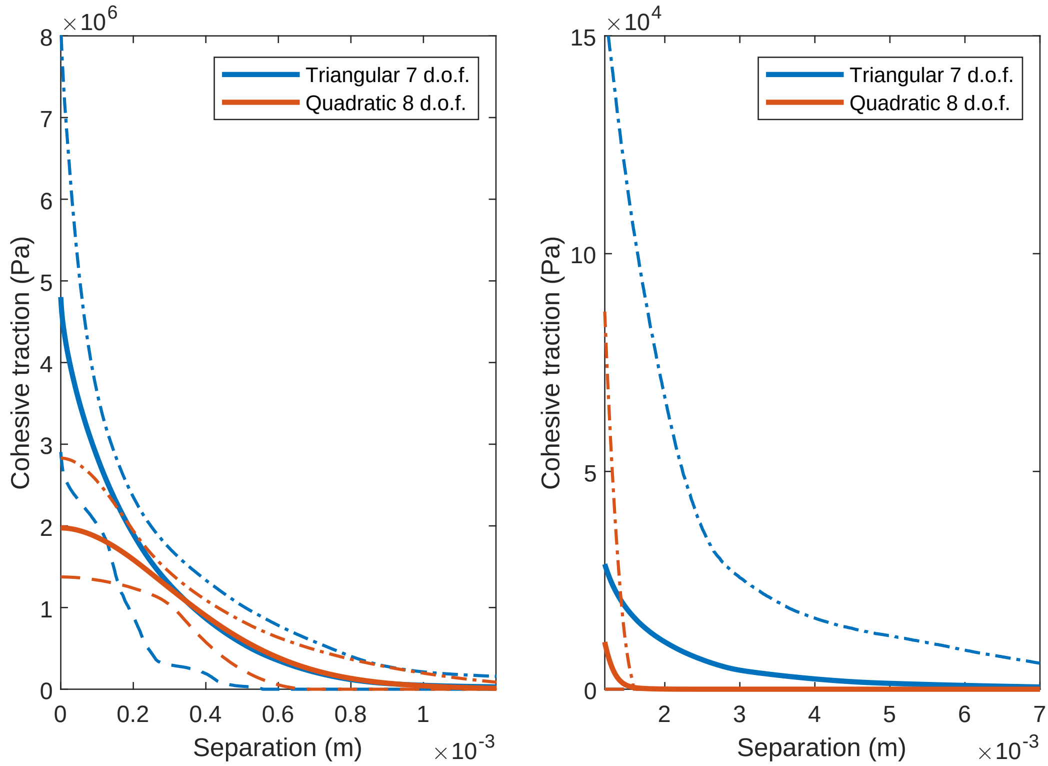


Figure 8.20: Resulting mean cohesive laws $[\delta_j, \overline{\sigma}_j^M]$ (continuous line) and confidence interval at 95% confidence level, delimited by the lower bound $\sigma^M_{LB,j}$ (dashed line) and the upper bound $\sigma^M_{LB,j}$ (dashed-dot line), obtained from the MC algorithm using 7 and 8 traction d.o.f. with the triangular loading functions and the quadratic loading functions, respectively. The legend shows the traction d.o.f. and the loading functions used to calculated each of the curves.

The results when all the pictures or only a single DIC picture is included show a high degree of resemblance.

From the little difference seen in between the results including a single picture and using all the pictures several conclusions can be obtained:

- The material has a high degree of homogeneity: the calculated cohesive law does not seem to have more uncertainty if it is calculated at different locations of the specimen or at a single location.
- The parameters driving the error are the uncertainties of the input parameters and the instability of the routine. The variation in displacement field used does not seem to affect the results.
- The experimental procedure has been preformed properly: the results does not seem to vary in between pictures which suggests a good quality of DIC results, where effects like out-of-plane movement have been avoided throughout the delamination.

9 | Conclusions

In this masters thesis an experimental procedure for characterization of cohesive laws of layered composite materials under delamination is designed with the aids of the Monte Carlo statistical approach for minimizing the influence of the experimental uncertainties on the results. The cohesive law calculations are done taking displacement fields using DIC as input to an inverse parameter identification algorithm that calculates cohesive tractions from the displacements measured (created in the previous semester project [Viejo et al., 2019]).

A 3D FE-model is developed using ANSYS Workbench to investigate the impact of the anticlastic bending on the input displacement, which has been demonstrated to not affect the results obtained for the case investigated. The DCB is modelled with the top part of the hinges to imitate the real behaviour of the specimen during the experiment, and a study of how the mispositioning of the hinges that might happen prior to performing the experiment affect the results obtained is conducted. Its demonstrated how a minimum rotation of 2º is needed for the cohesive laws predicted to have a significant deviation. A rotation of 2º can be easily spotted by the naked eye; thus, this issue is disregarded as it considered that it will not happen in reality.

The uncertainties of all variables that are involved in the experimental procedure are assessed and quantified and are to be used to study their influence on the variability of the cohesive laws calculated. For the variability study, the Monte Carlo approach is used, together with the gathered uncertainties and FE-Model displacements, to have information of the mean cohesive law and its confidence intervals. A correlation analysis is done on the variables involved in the Monte Carlo simulation using PPMCC and Distance Correlation to identify the variables that have more influence on the cohesive law confidence intervals. Flexural rigidity and shear rigidity are found to be the parameters driving the deviation on the calculated cohesive law bounds, and noise from DIC affects more the results when more traction d.o.f. are introduced.

To improve the robustness of the model, a study of the least-squares system is conducted to identify the best possible combination between number of traction d.o.f. and bounds for the location of the traction d.o.f, that produce the most robust system while assuring an accurate enough cohesive law prediction. The least-squares system is found to be highly sensitive to the solution found by the nonlinear optimizer, which calculates the optimum position of the traction d.o.f. to achieve the smallest residual norm, leading to an investigation focused on the MATLAB *fmincon* function parameters. Even though a good behaviour of the algorithm was observed, the solution quality did not improve, concluding that the triangular loading functions that described the shape of the interface tractions were too limited for this purpose.

Gathering all the information from the previous semester experimental procedure and the outcome of the Monte Carlo analyses, changes were done in the inverse parameter algorithm. Due to the negative effect of a wrong force measurement (which might happen in the delamination test), it is decided to include the force as and extra parameter of the least-squares system, being predicted by the inverse parameter routine. This change has been proved to predict the force applied in an accurate way and makes the experimental procedure much simpler as force does not have to be measured, in exchange of making the system slightly more sensitive to input noise. Triangular quadratic load functions have been introduced in the system to allow for a more flexible representation of the cohesive law shape, at a cost of introducing more traction d.o.f., which makes the least-squares procedure to have much more possible non-physical solutions. Constraints to the loading functions were introduced and proved to reduce the solution set of the least-squares system, improving the stability of the inverse parameter routine and the quality of the solution obtained, being able to represent better than before the material behaviour.

The performance of the new inverse parameter algorithm with the force, quadratic and triangular load functions, and constraints was tested with the aid of the Monte Carlo simulations and the uncertainties characterized for the variables. The improved inverse parameter routine proved to represent the cohesive law of the FE-model better and to reduce the resulting cohesive law confidence intervals. Correlation analyses were also conducted for the improved inverse parameter algorithm and the shear rigidity of the material was found to be the dominating variable related to the variability of the results, followed by the flexural rigidity of the material.

An experimental procedure was proposed, that solved the difficulties encountered on the previous semester project and reduces the impact of the variables that affect the most the accuracy and variability of the cohesive laws predicted (shear rigidity, flexural rigidity and DIC noise). The experiment proposed was performed using a Zwick machine to apply force to the hinges and cause the specimen to delaminate under pure mode I and the deflection of the DCB was measured using the DIC. The experiment was proved to be simple and fast to execute.

The experimental data was used as input for the inverse parameter formulation using triangular linear load functions and the triangular quadratic load functions with constraints to compare their performance. Both of them successfully predicted a similar cohesive law result for all the DIC displacements used as input. Finally, a Monte Carlo simulation was done where the samples generated take into account the variability of input displacements by including all the displacements from the experimental data set. None of the proposed loading function formulations that had been studied proved to have less variability and both models converged to the same solution with the number of traction d.o.f., which suggested the solution represents the behaviour of the material. However, discrepancies were observed on the shape of the cohesive laws predicted and are concluded to be a result of the different formulation of the load functions for each model.

The procedure followed to develop the inverse parameter identification tool is proven to be systematic and effective. The combination of FE-modelling responses to emulate experimental results, Monte Carlo simulations and correlation analyses provide a strong mechanism for design of experiments. Time-consuming experimental work is avoided through simulation and numerical modelling, identifying error sources and allowing for the evaluation of alternative procedures effectively and efficiently.

As overall conclusion for this master thesis, the methodology developed successfully characterizes the cohesive law shape of a material in a fast and an stable way when using real experimental data. However, the quality of the results has been demonstrated to strongly depend on the loading functions used to describe the interface tractions, and onset tractions and onset displacement values are not calculated consistently. The experimental procedure designed proves to be effective and easy to conduct, with a low amount of post processing required to obtain the data and just requiring the measurement of one variable.

10 Further Work

The resulting tool developed in this project relies on a theoretical background pertaining to a large amount of different fields. The depth achieved in each of them is limited due to lack of time. It is expected that considerable improvements can be achieved if further study is done in several aspects of the work done. They are listed in the following, classified in fields of application.

Formulation of the optimization problem

- Most of the data obtained by the DIC method is not used. Only the vertical displacement of the centerline is introduced in the inverse parameter identification routine. Including more information of the displacements and strains could improve the robustness and precision of the algorithm.
- Reduce the number of design variables used in the nonlinear optimization study to improve the robustness of the solution procedure.

Optimization procedure

- Deeper study of the influence of the initial guess and the convexity of the nonlinear optimization problem and its influence on the results. The relation in between the width of the confidence interval of the cohesive law and the performance of the nonlinear optimization routine needs from further investigation to ensure a robust solution procedure. This concerns including more parameters in the nonlinear optimization study and the investigation of other algorithms, e.g. genetic algorithms.

Beam subproblems

- Implementation of other beam theories of higher order may provide a more accurate representation of the real DCB problem.
- Other shapes of the load functions which provide a better representation of the real traction field.
- Implementation of other linear constraints.

Influence of other parameters over the results

- The shape of the cohesive law which is to be determined is expected to be strongly related to the performance of the inverse parameter identification routine. Further investigation of this topic would give more information about the limitations of the method and its

range of applicability. For example, preforming a correlation analysis of the Monte Carlo results of the experimental DIC measurements.
- Monte Carlo simulations together with correlation analyses have been proven to be a strong tool for identifying the parameters with most influence over the results. Only a very limited amount of parameters, the ones considered more important, have been included in this simulations. Parameters from the numerical routines as well as from the experimental procedure can be included to determine other sources of imprecision.

Design of experiments

- The correlation analyses performed provide an estimate of the influence of the values of each of the input parameters over the results. This information could be used to design the experiment where parameters like the geometry of the specimen can be chosen.

Experimental procedure

- The elastic properties of the DCB specimen have been shown to have a big impact on the results. Its mean values and uncertainties are obtained from external sources. Performing the experimental determination of these properties would allow for a more realistic assessment of the method.
- Perform a 3D DIC setup to evaluate the 3D effects of the real specimen.
- Synchronize the tensile test machine with the DIC camera to compare the force values calculated by the inverse parameter identification routine or to include them in the optimization problem.

The tool has shown a huge potential in the particular application studied here. The method developed relies on a concept with a very general applicability. Different possible expansions to other areas are displayed in the following.

- It can be applied to cyclic loading with no modification of the tool. Only the synchronization of the tensile test machine with the pictures needs to be done in order to relate the number of picture to the number of cycles.
- Including the axial deformation of the DCB allows the method to determine mixed mode I-II cohesive laws.
- Studying more complex configurations. The method relies on the combination of individual linear subproblems. Other analytic solutions different from a beam, or even Finite Element subproblems could be used.
- Study the possibility of using the confidence intervals obtained by MC for design purposes.

Bibliography

Aldrich, 1995. John Aldrich. *Correlation Genuine and Spurious in Pearson and Yule*. Statistical Science, 10(4), 364–376, 1995.

Arora, 2017. J.S. Arora. *Introduction to optimum design*. Elsevier, 2017.

Aster et al., 2013. Richard C. Aster, Brian Borchers and Clifford H. Thurber. *Parameter Estimation and Inverse Problems*. Acdemic Print Elsevier, 2013.

Atkinson, 1978. Kendall E. Atkinson. *An introduction to numerical analysis*. John Wiley and Sons, 1978.

Bai, 1992. Zhaojun Bai. *The CSD, GSVD, their Applications and Computations*, 1992. University of Minnesota.

Bak, 2015. Brian Lau Verndal Bak. *Progressive Damage Simulation of Laminates in Wind Turbine Blades under Quasistatic and Cyclic Loading*. PhD Thesis, 2015. Aalborg University.

Bak, 2017. Brian Lau Verndal Bak. *Introduction to Cohesive Zone Modeling*. Slides of PhD course, 2017. Aalborg University.

Barenblatt, 1962. G.I. Barenblatt. *The Mathematical Theory of Equilibrium Cracks in Brittle Fracture*. Advances in Applied Mechanics, 1962. Institute of Geology and Development of Combustible Minerals of the U.S.S.R. Academy of Sciences.

Beaurepaire and Schüeller, 2011. Pierre Beaurepaire and Gerhart Iwo Schüeller. *Modeling of the variability of fatigue crack growth using cohesive zone elements*. Engineering Fracture Mechanics, 78, 2399–2413, 2011.

Björck, 1996. Åke Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996. doi: 10.1137/1.9781611971484.

Calvetti et al., 2002. D. Calvetti, B. Lewis and L. Reichel. *GMRES, L-Curves, and Discrete Ill-Posed Problems*. BIT Numerical Mathematics, 42(1), 044–065, 2002.

Carter et al., 01 2014. Jennifer Carter, Michael Uchic and Michael Mills. *Impact of Speckle Pattern Parameters on DIC Strain Resolution Calculated from In-situ SEM Experiments*. Conference Proceedings of the Society for Experimental Mechanics Series, 66, 1–10, 2014. doi: 10.1007/978-3-319-06977-7_16.

Cook et al., 2002. Robert D. Cook, David S. Malkus, Michael E. Plesha and Robert J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley and Sons, Wisconsin, 2002.

DeCoursey, 2003. W.J. DeCoursey. *Statistics and Probability for Engineering Applications*. Elsevier, 2003.

Doyle, 2004. F. James Doyle. *Modern experimental analysis. Completing the solution of partially specified problems*. John Wiley & Sons, 2004.

Dugdale, 1960. D.S. Dugdale. *Yielding of steel sheets containing slits*. Mechanics of Materials, 1960. Engineering Department, University College of Swansea.

Elices et al., 2002. M. Elices, G.V Guinea, J. Gómez and J. Planas. *The Cohesive Zone Model: advantages, limitations and challenges*. Engineering Fracture Mechanics, 69, 137–163, 2002.

Golub and Kahan, 1965. G. Golub and W. Kahan. *Calculating the singular values and pseudo-inverse of a matrix*. SIAM Journal on Numerical Analysis, 2(2), 1965.

GOM mbH, 2015. GOM mbH. *GOM Correlate Professional, Inspection - 3D Testing*. Manual, 2015.

Goyal, 2003. Vinay K. Goyal. *Analytical Modeling of the Mechanics of Nucleation and Growth of Cracks*. Faculty of Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003.

Griffith, 1920. A. A. Griffith. *The phenomena of rupture and flow in solids*. Philos. Trans. Roy. Soc. London A, 221, 163–197, 1920.

Hadamard, 1902. J. Hadamard. *Sur les problemes aux dèrivèes partielles et leur signification physique*. Princeton University Bulletin, pages 49–52, 1902.

Hansen, 1998. P.C. Hansen. *Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial and Applied Mathematics (SIAM), 1998.

Heikkila and Silven, 1997. J. Heikkila and O. Silven. *A four-step camera calibration procedure with implicit image correction*. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997.

Hillerborg et al., 1976. Hillerborg, A. Modeer and M. Petersson. *Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements*. Cement and concrete research, 6, 773–782, 1976.

Irwin, 1957. G. R. Irwin. *Analysis of stresses and strains near the end of a crack traversing a plate*. Journal of Applied Mechanics, 24, 362–364, 1957.

JCGM-100, 2008. JCGM-100. *Evaluation of measurement data - Guide to the expression of uncertainty in measurement*, 2008. Joint Comitee for Guides in Metrology.

Jensen et al., 2019. Simon Mosbjerg Jensen, Mario Javier Martos, Esben Lindgaard and Brian Lau Verndal Bak. *Inverse parameter identification of n-segmented multilinear cohesive laws using parametric finite element modeling*. Composite Structures, 2019. Aalborg University.

Jones and Iadicola, 2018. E.M.C Jones and M.A. Iadicola. *A Good Practices Guide for Digital Image Correlation*. International Digital Image Correlation Society (iDICs), 2018. doi: 10.32720/idics/gpg.ed1.

Jones, 1999. Robert M. Jones. *Mechanics of composite materials*. Taylor and Francis, New York, 1999.

Joubert and Marwala, 2016. Daniel J. Joubert and Tshilidzi Marwala. *Monte Carlo Dynamically Weighted Importance Sampling for Finite Element Model Updating*. Topics in Modal Analysis & Testing, 10, 2016. Conference Proceedings of the Society for Experimental Mechanics Series. Springer, Cham.

King, 1989. J.E. King. *Failure in composite materials*. Metals and Materials, 5(12), 720–726, 1989.

Kirkup and Frenkel, 2006. L. Kirkup and R.B. Frenkel. *An introduction to uncertainty in measurement using the GUM*. Cambridge University Press, 2006.

Kroese et al., 2014. Dirk P. Kroese, Tim Brereton, Thomas Taimre and Zdravko I. Botev. *Why the Monte Carlo method is so important today*. WIREs Computational Statistics, 6, 386–392, 2014.

Lecompte et al., 08 2006. David Lecompte, Hugo Sol, John Vantomme and Anne Habraken. *Analysis of speckle patterns for deformation measurements by DIC*. Proceedings of SPIE - The International Society for Optical Engineering, 6341, 63410E–63410E, 2006. doi: 10.1117/12.695276.

Li and Ward, 1989. VC. Li and RJ. Ward. *A novel testing technique for post-peak tensile behaviour of cementitious materials.* Balkema Publishers, Rotterdam, 1989. in Fracture toughness and fracture energy–testing methods for concrete and rocks, by Mihashi H. and Takahashi H. and Wittmann FH.

Lindgaard, 2017. Esben Lindgaard. *Cohesive zone modelling with the Finite Element Method*. Slides of PhD course, 2017. Aalborg University.

Lindgaard and Bak, 2019. Esben Lindgaard and Brian Lau Verndal Bak. *Experimental characterization of delamination in off-axis GFRP laminates during mode I loading*. Composite Structures, pages 953–960, 2019.

Lund, 2019. Erik Lund. *Engineering Optimisation*. Slides of Masters course, 2019. Aalborg University.

Manshadi et al., 2014. B.D. Manshadi, E. Farmand-Ashtiani, J. Botsis and A.P. Vassilopoulos. *An iterative analytical/experimental study of bridging in delamination of the double cantilever beam specimen*. Composites: Part A, 2014.

MathWorks, 2019. Inc. The MathWorks. *MATLAB documentation*. `https://www.mathworks.com/help/matlab/index.html`, 2019.

Mosegaard and Tarantola, 1995. Klaus Mosegaard and Albert Tarantola. *Monte Carlo sampling of solutions to inverse problems*. Journal of Geophysical Research, 100(B7), 12431–12447, 1995.

Mouritsen, 2013. Ole Ø. Mouritsen. *Basic Uncertainty Evaluation by measurements using strain gauges*. Guide for students, 2013. Aalborg University.

Muttashar et al., 2015. M. Muttashar, W. Karunasena, A. Manallo and W. Lokuge. *Behaviour of hollow pultruded GFRP square beams with different shear span-to-depth rations*. Journal of Composite Materials, 2015.

Nocedal and Wright, 2006. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Science, 2006.

Ortiz and Pandolfi, 1999. M. Ortiz and A. Pandolfi. *Finite deformation irreversible cohesive elements for three-dimensional crack propagation analysis*. International journal of numerical methods in engineering, 44, 1267–1282, 1999.

Paige, 1984. C. C. Paige. *A note of a result of Sun Ji-Guang: sensitivity of the CS and GSV decompositions*. SIAM Journal on Numerical Analysis, 21(1), 1984.

Rabinovich, 2005. Semyon G. Rabinovich. *Measurement Errors and Uncertainties: Theory and Practice*. Springer, New Jersey, 2005.

Rice, June 1968. J.R. Rice. *A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks*. Journal of Applied Mechanics, 35, 379–386, 1968. Brown University.

Schmid, 1947. John Schmid. *The Relationship between the Coefficient of Correlation and the Angle Included between Regression Lines*. The Journal of Educational Research, 41(4), 311–313, 1947.

Shah et al., 1995. Surendra P. Shah, Stuart E. Swartz and Chengsheng Ouyang. *Fracture Mechanics of Concrete: Applications of Fracture Mechanics to Concrete, Rock and Other Quasi-Brittle Materials*. Wiley, New York, 1995.

Shen et al., 2010. Bin Shen, Ilinca Stanciulescu and Glaucio H. Paulino. *Inverse computation of cohesive fracture properties from displacement fields*. Inverse Problems in Science and Engineering, 2010.

Strang and Borre, 1997. Gilbert Strang and Kai Borre. *Linear Algebra, Geodesy and GPS*. Wellesley-Cambridge press, 1997.

Sun and Jin, 2012. C.T. Sun and Z.H. Jin. *Fracture Mechanics*. Elsevier, Aalborg, Denmark, 2012.

Székely et al., 2007. Gábor J. Székely, Maria L. Rizzo and Nail K. Bakirov. *Measuring and testing dependence by correlation of distances*. The Annals of Statistics, 35(6), 2769–2794, 2007. doi: 10.1214/009053607000000505.

Sørensen, 2010. Bent F. Sørensen. *Cohesive laws for assessment of materials failure: Theory, experimental methods and application*. Doctor of Technices thesis, 2010. National Laboratory for Sustainable Energy, Risø DTU.

Timoshenko, 1930. S. Timoshenko. *Strength of Materials. Part I*. D. Van Nostrand Company, 1930.

Timoshenko and Goodier, 1951. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. McGraw Hill, New York, 1951.

Turon et al., 2006. A. Turon, P.P. Camanho, J. Costa and G.C. Dávila. *A damage model for the simulation of delamination in advanced composites under variable-mode loading*. Mechanics of Materials, 2006.

Van Loan, 1976. Charles F. Van Loan. *Generalizing the Singular Value Decomposition*. SIAM Journal on Numerical Analysis, 13(1), 76–83, 1976.

Viejo et al., 2019. Félix Prieto Viejo, Pere J. Jaume Camps and José Mato Sanz. *Inverse Parameter Identification of Cohesive Laws through a Beam-Based Approach*. Third semester project, 2019. Aalborg University.

Waltz et al., 2006. R.A. Waltz, J.L. Morales, J. Nocedal and D. Orban. *An interior algorithm for nonlinear optimization that combines line search and trust region steps*. Mathematical Programming, 107, 391–408, 2006.

Yu and Lubineau, 2019. Lipin Yu and Guilles Lubineau. *Modeling of systematic errors in stereo-digital image correlation due to camera self-heating*. Scientific Reports, 9, 2019. doi: 10.1038/s41598-019-43019-7.

Zhang, 2000. Z. Zhang. *A flexible new technique for camera calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1330–1334, 2000.

A Cohesive Zone Modelling

In this appendix, the theoretical background, the constitutive, and kinematic relations of cohesive zone modelling are explained. Additionally, the implementation of this method onto the finite element environment is presented.

Based on the Damage continuum Mechanics framework, Cohesive Zone Modelling (CZM) arises as a tool which has the ability of removing the main problem that exists in Linear Elastic Fracture Mechanics (LEFM): the stress singularity at the crack tip. Additionally, its damage-based formulation allows the crack initiation study, which is not contemplated in LEFM, where an already existing crack is required. Another advantage is its implementation to the Finite Element Method, wherein studies such as crack growth evaluation can be simply performed.

In any composite structure experiencing a load, there exists a limit wherein the properties of the materials conforming the structure start to degrade gradually. This is known as damage. Stated earlier in the Introduction, delamination is one of the major problems for the structural integrity of composite materials, where the interfacial stiffness of the material diminishes as the delamination process goes on. When the delamination phenomenon is studied, there is a region of special interest. It is the one where the material interface is meant to break, denominated Fracture Process Zone (FPZ). Figure A.1 illustrates a double cantilever beam with forces applied at the end of its arms. The region highlighted in red, behind the crack tip, is the FPZ.

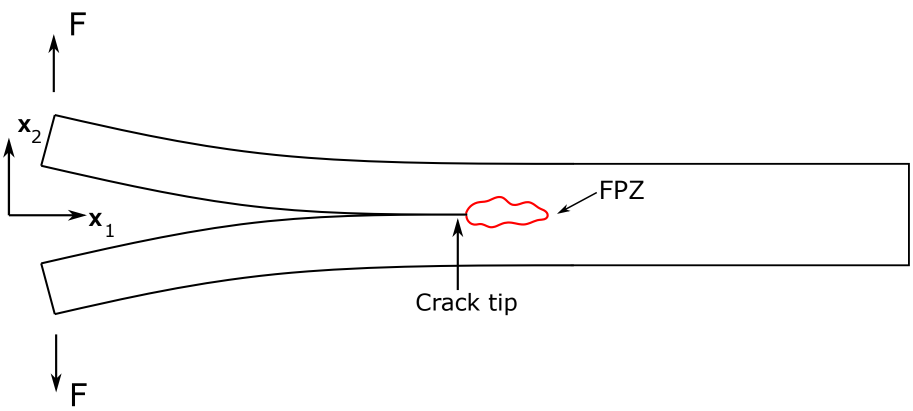


Figure A.1: Fracture process zone in a double cantilever beam with vertical forces applied.

A.1 The Cohesive Zone

In the middle of the 20th century, the foundations of the Cohesive Zone (CZ) were settled by Barenblatt [1962] and Dugdale [1960] separately, and from two different approaches. The CZ is defined as a region behind the crack tip where the interface is bonded as a result of the effect of a stress field, also known as cohesive tractions. The introduction of this concept has, as a consequence, that the singularity which is present in LEFM, is avoided i.e. the stresses are finite at the crack tip. Figure A.2 depicts an example of a traction field over a length L, acting at the interface of a DCB specimen under an axial loading state.

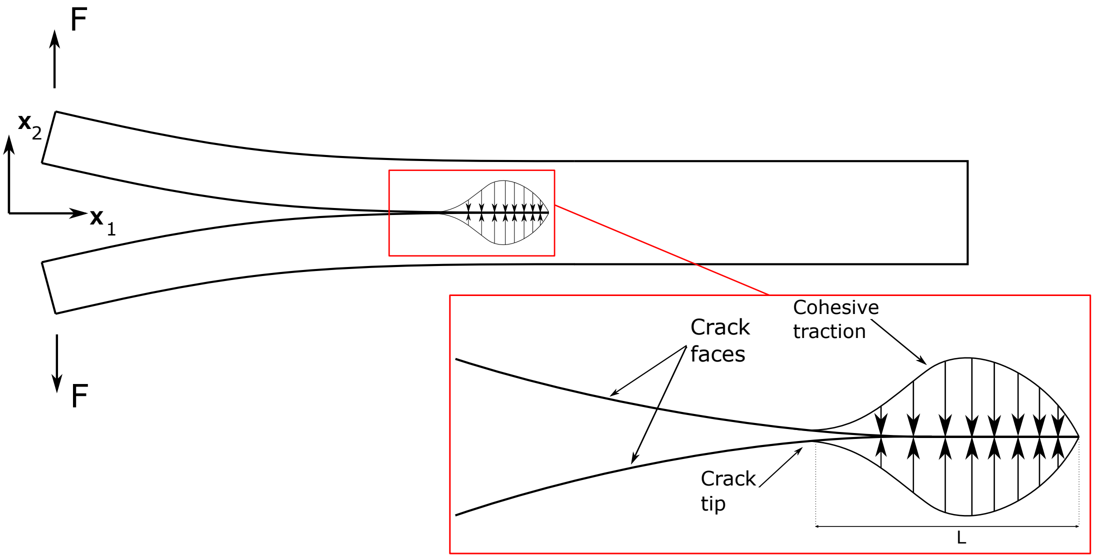


Figure A.2: Crack tip and the cohesive stresses acting at the cohesive zone.

A.1.1 Cohesive Law

When a determined load is applied onto a structure with an interface, the value of the stress at the interface τ, increases as it does the applied load. In the case of a pre-cracked specimen, as in Figures A.1 and A.2, once the stress at the crack tip reaches the value of the interface strength τ_c, damage starts. The faces of the crack start to separate, and new free areas are created progressively. Once the damage has started, the stress transmission decreases as the vertical displacement increases, see Figure A.3(a). The end point of this dynamic process is when the separation between faces δ, rises up to the critical displacement value δ_c, which is the same as saying that the point has suffered full damage.

At this point, it is of interest to establish a relation between the stress state and the vertical displacement of the interface. This is captured by the cohesive law; a constitutive relation which describes the response of the system when is subjected to a certain load state. It follows the expression below, where f is a dimensionless function describing the shape of the law [Sun and Jin, 2012].

$$\tau = \tau_c\, f(\delta/\delta_c) \tag{A.1}$$

An example of a bilinear cohesive law is presented in Figure A.3(b). Not all the laws have the same shape, as there can be trilinear, multilinear, exponential, etc. [Sun and Jin, 2012]. However, all of them have two regions in common, which are delimited by the onset traction τ_c, previously mentioned as interface strength, and the onset displacement δ_0. These regions are numbered as 1 and 2 in Figure A.3(b), and explained below.

1. Load carrying zone: from the free-of-load state ($\delta = 0$) to the onset displacement.

2. Damage zone: any displacement larger than δ_0 will cause an irreversible effect in the structure.

Following Sørensen [2010], cohesive laws are a material property; they are independent of the geometry of the structure. However, the value of the cohesive stresses is linked with the type of loading applied, thus by varying e.g. the loading rate on the same specimen, different shapes of cohesive law are obtained.

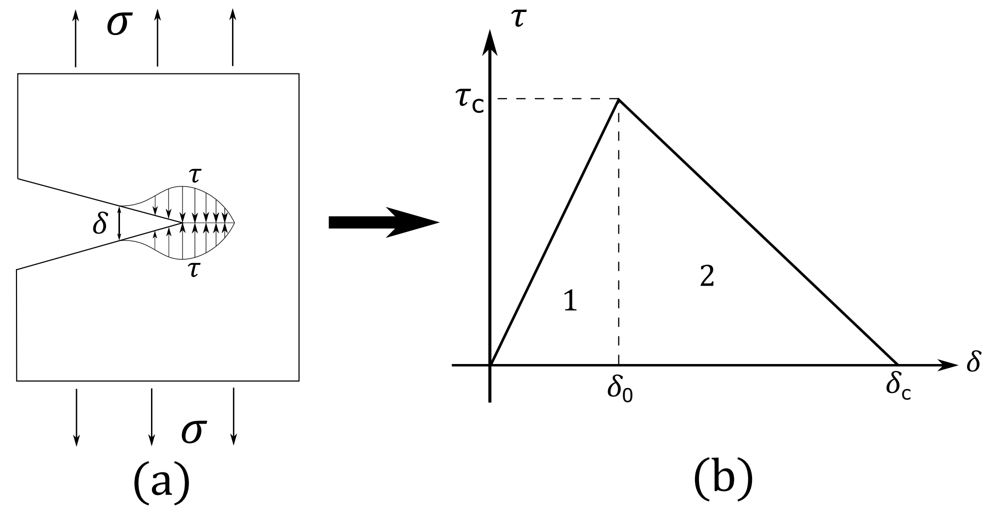


Figure A.3: (a) Cohesive tractions, τ, acting behind the crack tip. (b) Bilinear cohesive law.

The area under the cohesive law is defined as the cohesive energy density Γ_c, see Equation (A.2). It is demonstrated in Sun and Jin [2012] that the cohesive energy density equals the critical energy release rate G_c, a concept introduced in the LEFM environment that establishes a criterion for the evaluation of a crack.

$$\Gamma_c = \int_0^{\delta_c} \tau(\delta)\, d\delta \tag{A.2}$$

A.2 Boundary Value Problem and Interface Kinematics

In the following paragraphs, the governing equations of a crack propagation study are presented. In this case a discontinuous problem is faced, since to study the mechanics of the cohesive zone, an interface must be present. It is the presence of this interface and the inherent displacement jump between faces what makes the problem to be discontinuous. The following boundary value problem is then modified to comply with this requirements.

A.2.1 Boundary Value Problem

A body M with an existing crack such as the one depicted in Figure A.4 is considered. In the domain, different surfaces are defined. The crack itself is divided into two surfaces; the first, S_c, contains the entire crack, while the second, S_{coh}, comprises the region where the cohesive tractions are active. Without taking into account the crack region, the outer surfaces are separated as the one where the traction field is prescribed, S_T, and where the displacement field is prescribed, S_u. The interface S_d is modelled as a material discontinuity; meant to be the crack propagation front [Turon et al., 2006].

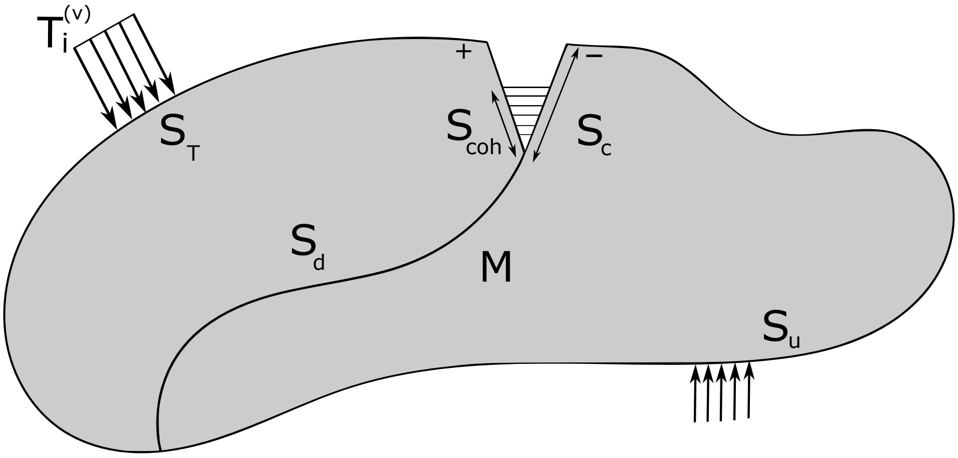


Figure A.4: Boundary value problem for a body with a discontinuity S_d.

Once the boundary conditions are defined, the stress field which acts inside the volume can be related to the external loads and tractions through the equilibrium equations, see Equations (A.3) through (A.5). The external loads are the volume forces, represented by B_i, and the surface tractions, $T_i^{(v)}$. The surface tractions are linked to the unit normal vector v_j through the Cauchy stress tensor σ_{ij}. The same expression is applied at the cohesive zone surface S_{coh}, as the normal vectors are related to cohesive stresses $\tau_i\pm$.

$$\sigma_{ij,j} + B_i = 0 \ \ in \ \ M \tag{A.3}$$

$$\sigma_{ij}v_j = T_i^{(v)} \ \ in \ \ S_T \tag{A.4}$$

$$\sigma_{ij}^+ v_j^+ = \tau_i^+ = \sigma_{ij}^- v_j^- = \tau_i^- \ \ in \ \ S_{coh} \tag{A.5}$$

A.2.2 Interface Kinematics

The boundary value problem presented in the previous paragraphs, introduced the interfacial surface S_d, where the crack propagation is expected to happen. Thus, the mechanics at the interface are of interest. What is desired at this point is to relate the global coordinate system, where the Equations of Elasticity are formulated, and the local coordinate system; where the cohesive law is defined. Once this relation is established, the BVP is closed. Figure A.5 displays the undeformed and the deformed configurations of the interface. The displacement jump $u_i^{\pm}$, is observed in the right-hand side of the illustration, and is defined in Equation (A.6).

$$[u_i] = u_i^+ - u_i^- \tag{A.6}$$

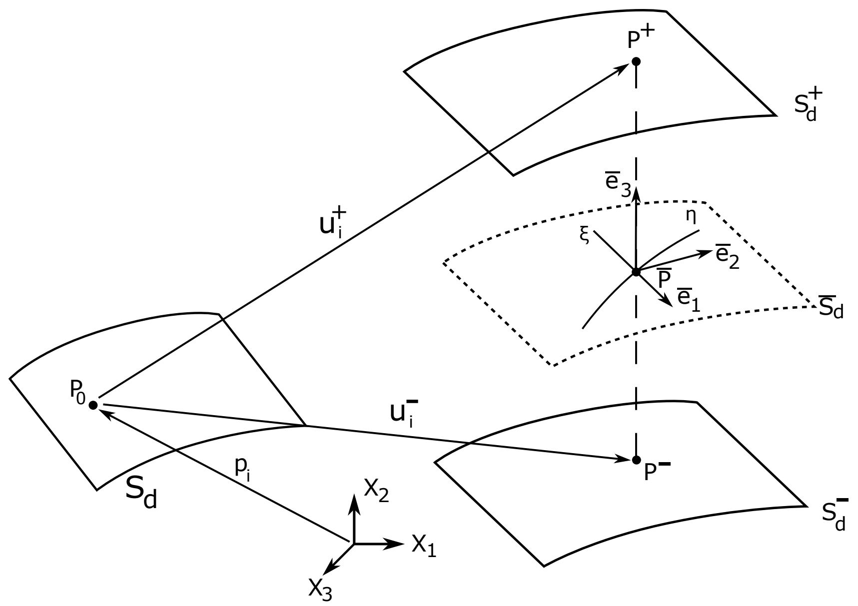


Figure A.5: Undeformed and deformed configurations of the interface with the global and local coordinate systems.

The displacement jump is described by the local Cartesian system $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$, which lays in the mid surface $\bar{S}_d$. This surface is allocated in the midpoint distance between points P^+ and P^-, which were held together before deformation, and its coordinates $\bar{x}_i$, are defined in the global coordinate system by means of Equation (A.7), wherein p_i states the position of any point P_0 in the undeformed interface.

$$\bar{x}_i = p_i + \frac{1}{2}(u_i^+ - u_i^-) \tag{A.7}$$

In order to obtain the local coordinate system, the vectors tangent to the midsurface at $\bar{P}$ are obtained as the gradients of the curvilinear coordinates of the mid-surface η and ξ. See Equation (A.8).

$$v_i^\eta = \frac{\partial \bar{x}_i}{\partial \eta} \qquad v_i^\xi = \frac{\partial \bar{x}_i}{\partial \xi} \tag{A.8}$$

These vectors are not always orthogonal, and the coordinate system would not be Cartesian. However, by the usage of cross products and the norms of the previously calculated vectors, the local Cartesian system can be calculated. From the unit vector of η, $\bar{e}_i^1$ is calculated as it follows.

$$\bar{e}_i^1 = \frac{v_i^\eta}{|v_i^\eta|} \tag{A.9}$$

Then, the mid-surface normal coordinate, $\bar{e}_i^3$ is obtained through the cross product of the two gradients, and then, it is normalized, see Equation (A.10).

$$\bar{e}_i^3 = \frac{v_i^\eta \times v_i^\xi}{\left|v_i^\eta \times v_i^\xi\right|} \tag{A.10}$$

The third component of the local coordinate system, $\bar{e}_i^2$ is calculated as the cross product of the already known values, $\bar{e}_i^1$ and $\bar{e}_i^3$.

$$\bar{e}_i^2 = \bar{e}_i^1 \times \bar{e}_i^3 \tag{A.11}$$

Finally, the rotation tensor Θ_{ij}, see Equation (A.12), can be established from the calculated local coordinate system, thus the interfacial vertical displacement can be expressed either in global or local coordinates. Following Bak [2015], the interface separation in terms of local coordinates, Δ_i, is expressed as in Equation (A.13)

$$\Theta_{ij} = \begin{bmatrix} \bar{e}_1^1 & \bar{e}_2^1 & \bar{e}_3^1 \\ \bar{e}_1^2 & \bar{e}_2^2 & \bar{e}_3^2 \\ \bar{e}_1^3 & \bar{e}_2^3 & \bar{e}_3^3 \end{bmatrix} \tag{A.12}$$

$$\Delta_i = \Theta_{ij}(u_i^+ - u_i^-) \tag{A.13}$$

A.3 Finite Element Implementation

The set of expressions presented through Equations (A.3) to (A.5) are the equations of Elasticity, which must be derived in order to obtain a solution for the boundary value problem, which in this case, presents a discontinuous system to be solved. As the governing equations are partial differential equations, it is said that the problem is written in strong form [Cook et al., 2002]. For complex geometries, the problem becomes almost impossible to solve. The Finite Element Method (FEM) appears as a mechanism with the ability to solve the partial differential equations, by dividing the whole system into smaller subsystems, denominated elements. The whole compound of subsystems is then a system of a finite number of degrees of freedom (d.o.f.), and is termed a mesh. This process is called discretization, and allows the formulation of the partial differential equations as a set of linear algebraic equations, which are handily solved.

Moving onto Cohesive Zone Modelling, its formulation requires of an element which is capable to deal with the interface relations. The following paragraphs are based on Lindgaard [2017]

A.3.1 Interface Element

Taking as a departure point the definition of the interface kinematics, the cohesive zone element can be developed. The elements present at the cohesive zone are defined in the same manner as the interface elements, which are formulated as an isoparametric element with 8 nodes. When the faces are held together, the element has zero thickness, but when the displacement jump is active, the element faces start to separate, thus gaining thickness. See Figure A.6.

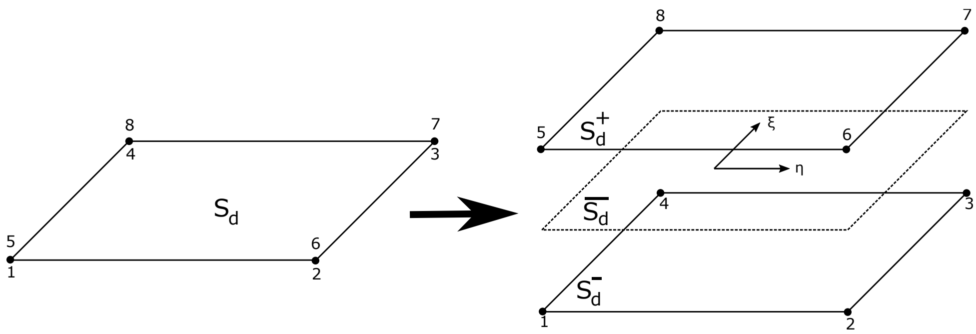


Figure A.6: Undeformed and deformed configuration of the interface element.

Isoparametric formulation is characterized by using the same shape functions for the geometry and for the displacements. This formulation introduces a local system of coordinates (η,ξ), so the element is mapped as a square element. Shape functions, and the shape function matrix are defined in the following expressions, which are written in matrix form.

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \\
N_2 &= \frac{1}{4}(1+\xi)(1-\eta) \\
N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \\
N_4 &= \frac{1}{4}(1-\xi)(1+\eta)
\end{aligned}
\tag{A.14}
$$

$$
[N]^+ = \begin{bmatrix} N_1 & 0 & 0 & & N_4 & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_4 & 0 \\ 0 & 0 & N_1 & & 0 & 0 & N_4 \end{bmatrix} \tag{A.15}
$$

$$
[N]^- = -[N]^+ \tag{A.16}
$$

$$
[N] = \big[[N]^-, [N]^+\big] \tag{A.17}
$$

The superscripts in the equations are to distinguish the different surfaces of the crack. The ones containing $+$ are the upper face, while the others, with $-$, are the lower face of the interface. The separation in the element is interpolated bilinearly with the expression below, where $\{d\}$ are the nodal displacements

$$
\{u\}^+ - \{u\}^- = [N]\{d\} \tag{A.18}
$$

Then, to obtain the values in the global coordinate system, the rotation matrix derived in Equation (A.12) is introduced, and the vertical displacement of the deformed surface can be calculated.

$$
\{\Delta\} = [\Theta][N]\{d\} \tag{A.19}
$$

A.3.2 Principle of Virtual Work

The principle of virtual work, also known as the principle of virtual displacements, is a tool which is used in the FEM to implement the governing equations. Due to its applicability to nonlinear problems, it is well-suited for the derivation of the tangent stiffness matrix and the internal force vector in a boundary value problem with an interface, presented before. In the principle of virtual work, the premise which must be fulfilled is that when a small or virtual variation δ, in the displacements or strains is introduced to the system, the work done by the internal and the external forces is in equilibrium. Equation (A.20) represents the virtual work of internal and external loads for an elastic continuum, where $\{\varepsilon\}$ is the strain tensor, $\{\sigma\}$ is the Cauchy stress tensor, $\{u\}$ is the displacement field, $\{F\}$ are the volume forces, $\{\Phi\}$ are the surface tractions, and $\{p\}$ are the point forces.

$$\int_V \{\delta\varepsilon\}^T\{\sigma\}dV = \int_V \{\delta u\}^T\{F\}dV + \int_S \{\delta u\}^T\{\Phi\}dS + \sum_{i=1}^{n}\{\delta u\}_i^T\{p\}_i \tag{A.20}$$

The fact that this formulation deals with discontinuous systems i.e. the interface, implies that the formulation will have an extra term accounting for the work done at the interface [Goyal, 2003]. This can be seen in Equation (A.21), where the second integral is the work done by the interface tractions. The rotation tensor is $[\Theta]$, and the interface stresses, $\{\delta\tau\}$.

$$\int_V \{\delta\varepsilon\}^T\{\sigma\}dV + \int_{\overline{S}} \delta\left\{\{u\}^+ - \{u\}^-\right\}^T[\Theta]^T\{\tau\}d\overline{S} =$$
$$\int_V \{\delta u\}^T\{F\}dV + \int_S \{\delta u\}^T\{\Phi\}dS + \sum_{i=1}^{n}\{\delta u\}_i^T\{p\}_i \tag{A.21}$$

If the second integral from the above equation is taken, and the relation in Equation (A.18) is introduced, the expression for the internal work of an interface element is the following.

$$\delta W_{int} = \int_{\overline{S}} \delta\{d\}^T[N]^T[\Theta]^T\{\tau\}d\overline{S} \tag{A.22}$$

Then, the variation of the displacement field is taken out from the integral, and what is left inside is known as the internal force vector, r_{int}.

$$\delta W_{int} = \delta\{d\}^T \int_{\overline{S}} [N]^T[\Theta]^T\{\tau\}d\overline{S} = \delta\{d\}^T\{r_{int}\} \tag{A.23}$$

$$\{r_{int}\} = \int_{\overline{S}} [N]^T[\Theta]^T\{\tau\}d\overline{S} \tag{A.24}$$

The integration limits of the previous expressions are taken from a global reference system, but when it comes to evaluate the internal forces, the coordinate system used in the integration bounds must be the local one. For that, the Jacobian is introduced, as it enables the relation of the deformed mid-surfaces from local ($d\eta\ d\xi$) to global ($dx\ dy$) limits, such as

$$\left\{r^{int}\right\} = \int_{\overline{S}} [N]^T[\Theta]^T\{\tau\}d\overline{S} = \int_\xi \int_\eta [N]^T[\Theta]^T\{\tau\}Jd\xi d\eta \tag{A.25}$$

where the Jacobian, J, is defined as

$$J = \left|v^\xi \times v^\eta\right| \tag{A.26}$$

The tangent stiffness matrix, $[k_T]$ is then identified as the first variation of the internal force vector.

$$\delta\{r_{int}\} = [k_T]\delta\{d\} \tag{A.27}$$

Then, by taking the first variation of the internal force vector (Equation (A.25)), and using the chain rule, the following is obtained.

$$\delta\left\{r^{int}\right\} = \int_\xi\int_\eta \left([N]^T\delta[\Theta]^T\{\tau\}J + [N]^T[\Theta]^T\delta\{\tau\}J + [N]^T[\Theta]^T\{\tau\}\delta J\right) d\xi d\eta \tag{A.28}$$

The variations of the geometric parameters (Θ and J) are usually neglected. Thus, the first and third terms of the integral are disregarded, leaving the expression below.

$$\delta\left\{r^{int}\right\} \approx \int_\xi\int_\eta [N]^T[\Theta]^T\delta\{\tau\}Jd\xi d\eta \tag{A.29}$$

The variation of the stresses at the interface can be substituted by the constitutive tangent stiffness tensor D_{tan}, which relates the variation of the tractions with the variations of the local displacement vector [Turon et al., 2006]. Therefore, this relation is introduced into the expression for the variation of the internal force vector, see Equations (A.30) and (A.31).

$$\delta\{\tau\} = [D_{tan}]\delta d \tag{A.30}$$

$$\delta\left\{r^{int}\right\} \approx \int_\xi\int_\eta [N]^T[\Theta]^T\left[D_{tan}\right][\Theta][N]\delta\{d\}Jd\xi d\eta \tag{A.31}$$

Finally, the tangent stiffness matrix is the expression left when the variation of displacements is taken out from the integral.

$$\delta\left\{r^{int}\right\} \approx [k_T]\,\delta\{d\} \tag{A.32}$$

$$[k_T] \approx \int_\xi\int_\eta [N]^T[\Theta]^T\left[D_{tan}\right][\Theta][N]Jd\xi d\eta \tag{A.33}$$

Both the internal force vector and the stiffness matrix can be calculated by means of numerical integration rules e.g. the Gauss quadrature or the Newton-Cotes method.

Once the expressions for the stiffness matrix and the force vector of a single element have been derived, the contribution of all the elements present at the interface, n_e, is added, so global values are obtained. The following expression defines the global stiffness matrix, and the global internal and external force vectors, in this order.

$$[K_T] = \sum_{n=1}^{n_e}[k_T], \quad \left\{R^{int}\right\} = \sum_{n=1}^{n_e}\left\{r^{int}\right\}, \quad \left\{R^{ext}\right\} = \sum_{n=1}^{n_e}\left\{r^{ext}\right\} \tag{A.34}$$

Since the constitutive relation of the cohesive elements is nonlinear, the linear dependency of the nodal displacements and internal forces is no longer present. Therefore, equilibrium is formulated as a residual R between the internal and the external forces, see Equation (A.35).

$$\{R\} = \left\{R^{int}\right\} - \left\{R^{ext}\right\} = \{0\} \tag{A.35}$$

If it is assumed that the load is applied in steps such as $n = 1, 2, ..$, and that the exact solution of the nodal displacement vector is accurately approximated by $\{D\}_i^n$, the residual of the next iteration can be approximated by a first order Taylor expansion.

$$\left\{R\left(\{D\}_{i+1}^n\right)\right\} \approx \left\{R\left(\{D\}_i^n\right)\right\} + \frac{\partial\left\{R\left(\{D\}_i\right)\right\}}{\partial\{D\}}\delta\{D\}_i^n = \{0\} \tag{A.36}$$

The thid term of Equation (A.36) is the tangent stiffness matrix evaluated at the current iteration step. Then, the incremental equilibrium equation is written as follows. As the equation is solved with respect to the variation of the global displacement vector, the iterations are updated with this value i.e. $\{D\}_{i+1}^{n} = \{D\}_{i}^{n} + \delta\{D\}_{i}^{n}$.

$$[K_T(\{D\}_i^n)]\,\delta\{D\}_i^n = -\,\{R(\{D\}_i^n)\} \tag{A.37}$$

B Inverse Problems using Least-Squares

The present appendix describes the use of linear regression for solving overdetermined inverse problems. The ill-conditioning of a system is defined and the tools for identifying an ill-conditioned system are presented. An overview of the sensitivity of the solution of the system to input noise is given. Finally, regularization priciples are shown to attenuate the effect of ill-conditioning of the system.

B.1 Inverse problems in Engineering

This chapter is based on Aster et al. [2013].

Inverse problems have their natural origin if one is interested in determining the internal structure of a physical system from the system's behaviour [Hansen, 1998]. It is common in the scientific area, that engineers or scientists wish to relate physical parameters that characterize a model response m and observations from experiments registered and saved as a set of data d. Thus, considering a discrete number of observations d, a relation between both of them might be established in vector-matrix form such as

$$Gm = d \tag{B.1}$$

Where d is considered as a set of discrete observations and G can be seen as any type of operator. Real data d_{true} is always perturbed by a certain amount of noise η, that does not exactly match the underlying real response m_{true} of the system. Therefore, the collection of data d can be seen as a combination of real data d_{true} plus a noise component $d = d_{true} + \eta$. The goal of an inverse method is to recover the real response of the model m_{true} from the set of perturbed data d [Aster et al., 2013].

$$d = G\left(m_{\text{true}}\right) + \eta = d_{\text{true}} + \eta \tag{B.2}$$

In some situations, a small noise contribution η can make the obtained response m have little or no correspondence at all with m_{true} [Aster et al., 2013]. Usually, when trying to determine the model response m from experimental data, the number of observations d recorded exceeds the number of parameters that are being calculated m, which means that the system is *over-determined*.

Over-determined systems are most of the times inconsistent, thus, they have no solution. No solution exists because the noisy data d lays usually out of the range of coefficient matrix G, meaning that with the linear system of equations the real value of the response m_{true} is never obtained. This is the reason why instead of looking for the exact solution of the system m_{true}, an approximation of the real solution m that minimizes some measure of the misfit between data and model is computed [Aster et al., 2013].

B.2 Linear Regression and Least-Squares

The problem of finding a curve to approximately fit a given set of data is denoted as regression. As mentioned in the previous paragraph, a measure of mismatch between the data must be found, to minimize it and obtain an approximate solution of the system that is needed to be solved. This measure is the residual r between model predictions and data recorded.

$$r = Gm - d \tag{B.3}$$

The most common way of measuring the magnitude of the residual is to use the *"2-norm or Euclidean norm"* of the residual vector, which is denominated as *least-squares solution*. This norm calculates the length of a vector as $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$. Thus, the problem turns into an unconstrained optimization problem where the aim is to minimize the norm of the residual, to get the best solution possible for the data fit.

$$\text{minimize } \|r\| = \|Gm - d\|_2^2 = \sum_{i=1}^{k} \left(G_i^T m - d_i\right)^2 \tag{B.4}$$

Where $G \in \mathbb{R}^{k\times n}$(with $k \geq n$), G_i^T are the rows of G and the vector $m \in R^n$ is the model response. The least-squares solution can be reduced to solving a set of linear equations such as,

$$\left(G^T G\right) m = G^T d \tag{B.5}$$

The analytical solution of the least-squares (Ordinary Least Squares solution) problem to the set of linear equations above is

$$m = \left(G^T G\right)^{-1} G^T d \tag{B.6}$$

B.3 Ill-Conditioning of a System Matrix

A system of linear equations can be denoted as well-conditioned or *ill-conditioned* depending on its properties. The concept of ill-conditioning was first introduced by Hadamard [1902] defining it as *"a system with multiple solutions or that is not a continuous function of the data."*

This effect can be seen when a small noise variation causes the system to become highly unstable, providing an undesired solution [Aster et al., 2013]. Note that "undesired solution" means that the results obtained are numerically good but not necessarily physically plausible e.g. a highly oscillatory solution.

Ill-conditioning is not a property of the data itself, but it is related to the system, which in this case, following the notation introduced in previous section, is a property of the matrix G [Aster et al., 2013]. Therefore, a set of measurements d with a certain degree of noise η help in the determination of the degree of ill-conditioning that the system has on its formulation.

B.3.1 Singular-Value Decomposition (SVD)

One of the most essential tools for analysing ill-conditioned and rank deficient problems is the Singular Value Decomposition (SVD) [Hansen, 1998]. If matrix $G \in \mathbb{R}^{k\times n}$ previously introduced is a rectangular or square matrix, then the SVD decomposition of G takes the form

$$G = U\Theta V^T = \sum_{i=1}^{n} u_i \theta_i v_i^T \tag{B.7}$$

where $U = (u_1, \ldots, u_k) \in \mathbb{R}^{k\times k}$ and $V = (v_1, \ldots, v_n) \in \mathbb{R}^{n\times n}$ have orthonormal columns and the matrix $\Theta \in \mathbb{R}^{k\times n}$ is a diagonal matrix with non-negative diagonal elements appearing in nonincreasing order

$$\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n \geq 0 \tag{B.8}$$

These values θ_i are denoted as *singular values*, which are key for characterizing the type of problem to be analysed. Moreover, vectors u_i and v_i are named left and right singular vectors, respectively. With the SVD decomposition, the solution for the Least-Squares problem presented in Equation (B.4) can be expressed as

$$m_{\text{LS}} = \sum_{i=1}^{\text{rank}(G)} \frac{u_i^T d}{\theta_i} v_i \tag{B.9}$$

By observing the left and right singular vectors u_i and v_i, one can also identify ill-conditioning when their components tend to have more significant sign changes as index i increases. Thus, as θ_i decreases vectors u_i and v_i become more oscillatory [Hansen, 1998].

There are two important classes of ill-conditioned problems according to the properties of the coefficient matrix G:

1. **Rank-deficient** problems are characterized by having a coefficient matrix presenting a big difference between big and small singular values. This means one or more columns and rows of the coefficient matrix G are almost linear combinations of some or all of the rest of rows and columns. Thus, matrix G contains redundant information that has to be eliminated [Hansen, 1998]. In this case, singular values that are small are to be identified and "ignored", replacing the remaining singular values for their reciprocal, creating an approximation of the original matrix that is better conditioned [Golub and Kahan, 1965]. However, none of the cases studied in this project belong to this classification of problems.
2. **Discrete ill-posed** problems are a result of discretizing ill-posed continuous problems. Ill-posed problems arise e.g. when trying to determine the internal behaviour of a system based on external measurements [Calvetti et al., 2002].The singular values of matrix G decay gradually to zero without a big gap between adjacent singular values. These systems are assumed to be inconsistent and often arise from seeking to obtain a minimal

norm solution from a system like Equation (B.1) [Calvetti et al., 2002]. The way to deal with these systems, is to find a balance between the residual norm and the solution that satisfies the user's expectation, which is the principle of regularization techniques [Hansen, 1998]. All the systems involved in this project belong to this classification of problems.

In the case that the coefficient matrix G is ill-conditioned but does not belong to either one of the two groups presented above, regularization is not applicable to obtain a suitable solution. Then, the problem is to be solved as accurately as possible by other methods [Hansen, 1998].

B.3.2 Generalized Singular-Value Decomposition (GSVD)

Note that in this section, same symbols as in the SVD are used to define different matrices e.g. U, V and Θ. However, in this project it is always mentioned which decomposition is used for the calculations.

The GSVD of a matrix pair (G,Z), was first introduced by Van Loan [1976]. It is defined as, literal from Hansen [1998], *"a generalisation of the SVD of G in the sense that the generalized singular values of the matrix pair are essentially the square roots of the generalized eigenvalues of the matrix pair G^TG, Z^TZ"*. Dimensions of G and Z are assumed to be $G \in \mathbb{R}^{k\times n}$ and $Z \in \mathbb{R}^{p\times n}$ where $k \geq n \geq p$. It is also assumed that the null space of both matrices is the same and that Z has full row rank, the GSVD decomposition is done as follows

$$G = U\Theta X^{-1}, \quad Z = VMX^{-1} \tag{B.10}$$

Where $U \in \mathbb{R}^{k\times n}$, $V \in \mathbb{R}^{p\times p}$ and $X \in \mathbb{R}^{n\times n}$, then Θ and M are diagonal $p \times p$ matrices which contain nonnegative values ordered such that

$$0 \leq \theta_1 \leq \cdots \leq \theta_p \leq 1, \quad 1 \geq \mu_1 \geq \cdots \geq \mu_p > 0 \tag{B.11}$$

and they are normalized as

$$\theta_i^2 + \mu_i^2 = 1, \quad i = 1, \ldots, p \tag{B.12}$$

The values related represent the generalized singular values γ_i of the pair of matrices G, Z.

$$\gamma_i = \theta_i/\mu_i, \quad i = 1, \ldots, p \tag{B.13}$$

Note that the generalized singular values appear now in nondecreasing order, opposite to the particular singular values (when describer in previous section for the SVD), due to historical reasons [Hansen, 1998]. For a discrete ill-posed problem, the following properties of the singular values for the GSVD, similar to those explained in the previous section for the SVD, are found:

1. The generalized singular values γ_i decay to zero with no gap in the set as shown in Figure B.1. When the dimensions of matrix G are greater there is an increase in the number of small singular values.

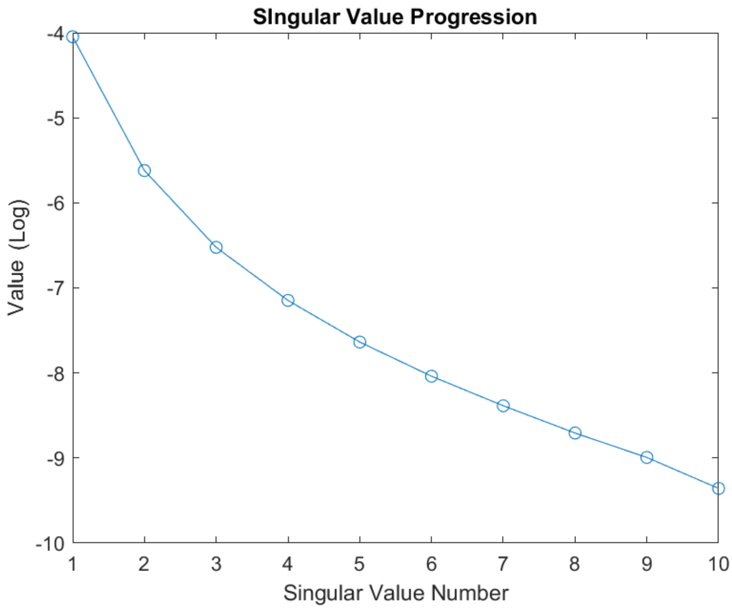


Figure B.1: Singular values tending to zero as index i increases.

2. The components of the singular vectors u_i, v_i and x_i corresponding to the the columns of matrices U, V and X have more sign changes as index i increases. An example of this behaviour is shown in Figure B.2.

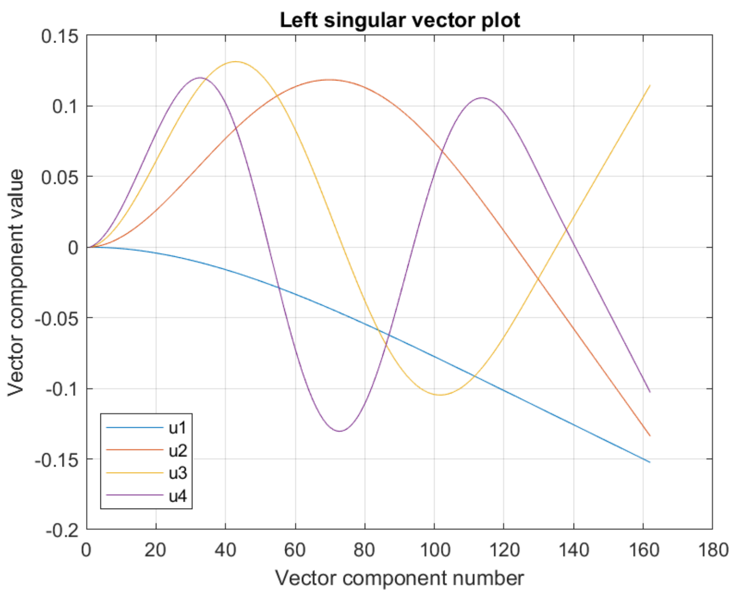


Figure B.2: Effect of the index i on singular vector component values.

This decomposition can be used when applying regularization to the system. However, in this project it is exclusively used to solve the Linear Least Squares system with Equality Constraints, explained in the following section.

B.3.3 Linear Least-Squares with Equality Constraints

This section is based on Golub and Kahan [1965] and Björck [1996].

The introduction of linear equality constraints in the least squares has its origin on the weighted least squares where observations with larger weights influence the solution more than those with lower weights. Therefore, this principle implies that observations with large weights can, indeed, act as linear constraints [Strang and Borre, 1997].

The starting point of the problem is to have a linear system that is needed to be minimized using linear least squares. This system is subjected to a set of constraints that come (in this particular case) from a physical model. Then, the problem becomes a *Least-Squares Problem with Linear Equality Constraints*, that has the form:

$$\min_{m} \|Gm - d\|^2 \quad \text{subject to} \quad Zm = b \tag{B.14}$$

This problem can be solved using the GSVD explained in the previous section, by using the transformations of matrices and their properties to obtain a solution to the problem in Equation (B.14).

$$\min_{m} \left\|\Theta X^{\mathrm{T}}\boldsymbol{m} - U^{\mathrm{T}}\boldsymbol{d}\right\| \quad \text{subject to} \quad MX^{\mathrm{T}}\boldsymbol{m} = V^{\mathrm{T}}\boldsymbol{b} \tag{B.15}$$

With a column oriented notation of the group of matrices U, V and X, then

$$\left(X^{\mathrm{T}}\boldsymbol{m}\right)_i = \begin{cases} \frac{v_i^{\mathrm{T}}b}{\mu_i} & \text{for } i = 1, \ldots, p \\ \frac{u_i^{\mathrm{T}}d}{\theta_i} & \text{for } i = p+1, \ldots, n \end{cases} \tag{B.16}$$

The solution obtained from the linear least squares with linear constraints using GSVD is

$$\boldsymbol{m}_c = \sum_{i=1}^{p} \frac{\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{b}}{\mu_i}(X^{\mathrm{T}})^{-1} + \sum_{i=p+1}^{n} \frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{d}}{\theta_i}(X^{\mathrm{T}})^{-1} \tag{B.17}$$

B.3.4 Noise Magnification and the Condition Number

If a mapping of a random vector z onto G is considered using the SVD

$$Gz = \sum_{i=1}^{n} \theta_i \left(v_i^T z \right) u_i \tag{B.18}$$

It can be seen how the multiplication by θ_i, when θ_i decreasing to zero (due to ill-conditioning), makes the response x components with a higher i index be more damped the rest. Therefore, the low number of oscillations seen on the left and right singular vectors u_i and v_i with smaller i dominate the solution. However, when computing the inverse problem (solution shown in Equation (B.9)), the opposite effect happens. The high frequency oscillations of the observations b get amplified, transmitting the oscillatory behaviour to the solution calculated m. Therefore, errors committed in the measurements, are magnified by the system, giving an incorrect solution [Hansen, 1998]. Taking the Least-Squares solution using SVD in Equation B.9, when a set of observations d is seen as a real value d_{true} plus a noise component η the equation becomes

$$m_{\mathrm{LS}} = \sum_{i=1}^{\mathrm{rank}(G)} \frac{u_i^T (d + \eta)}{\theta_i} v_i \tag{B.19}$$

It is seen, how even if the noise component η is small due to a small error in the measurements done, when θ tends to zero the whole expression $\frac{u_i^T \eta}{\theta_i}$ can play an important part on the solution obtained. This example reinforces the importance of the magnification of the errors in the input by the characteristics of the matrix G of the system.

The sensitivity of the solution m of the system due to a small perturbation η on vector d can be quantified by the study of the *condition number* of the coefficient matrix G, denoted as $cond(G)$. The condition number relates the influence of the relative error of the vector of discrete observations

$$\|d_{\mathrm{true}} - d\| / \|d\|$$

on the relative error of the solution

$$\|m_{\mathrm{true}} - m\| / \|m\|$$

, which is also an indicator of ill-conditioning of the system [Atkinson, 1978]. The objective of the condition number is to be used for the computation of error bounds, having

$$\frac{\|\eta\|}{\|G\|\|m\|} \leq \frac{\|e\|}{\|m\|} \leq \frac{\|G^{-1}\| \, \|\eta\|}{\|m\|} \tag{B.20}$$

where $\eta = d_{\mathrm{true}} - d$ and $e = m_{\mathrm{true}} - m$, and with the bounds

$$\|d\| \leq \|G\|\|m\| \quad \|m\| \leq \left\| G^{-1} \right\| \|d\| \tag{B.21}$$

the following solution is obtained for the error bound

$$\frac{1}{\|G\| \, \|G^{-1}\|} \cdot \frac{\|\eta\|}{\|d\|} \leq \frac{\|e\|}{\|m\|} \leq \|G\| \left\|G^{-1}\right\| \cdot \frac{\|\eta\|}{\|d\|} \tag{B.22}$$

where $\|G\|\|G^{-1}\| = cond(G)$ is the condition number. Note that for an *overdetermined system*, the computation of the condition number of the coefficient matrix G is done by calculating the ratio between its biggest and smallest nonzero singular values $\theta_1/\theta_{\mathrm{rank}(G)}$.

The condition number depends on the norm that is being used and it is always bounded below by one such that

$$1 \leq \|I\| = \left\|GG^{-1}\right\| \leq \|G\| \left\|G^{-1}\right\| = \mathrm{cond}(G) \tag{B.23}$$

If the condition number is large, the error bounds become excessive, and the system is highly unstable [Atkinson, 1978]. It is difficult to define a quantity from which the condition number can be considered as large, as it is problem dependent. For this project, to see which system is better conditioned, the evolution of the condition number as changes are introduced is evaluated, rather than the quantity itself.

For the sensitivity of the computations when using the GSVD method, the considerations differ. For this situation, being the condition number of a rectangular matrix defined as the ratio between the biggest and smallest singular value $\theta_1/\theta_{\mathrm{rank}(G)}$ the quantity of the singular value when there is a perturbation on each of the input matrices is important. Numerical analysts have proven that having a small perturbation for matrices G and Z does not mean that the GSVD matrix pairs also have a small change in coefficients [Paige, 1984]. This is essentially due to the fact that the matrix X, introduced in equation (B.10), can be a poorly condition matrix [Bai, 1992]. Therefore, the focus when analysing the sensitivity of the system solved by GSVD is on the conditioning of matrix X. Observing the solution of the system using the GSVD (see Equation (B.17)), both factors are multiplied by the inverse of X^T, which makes the error bound to be dependent on the condition number of matrix X.

B.3.5 Regularization and the L-Curve Approach

Depending on the outcome of the singular value analysis, the treatment for the ill-conditioning varies. When the system is proven to be a discrete ill-posed system a technique denoted as *regularization* can be used to deal with solution instabilities.

The use of regularization involves incorporating further information about a desired solution in order to stabilize the problem and obtain a single and useful solution [Doyle, 2004]. Regularization involves the computation of a residual, associated with the regularized solution, and the application of four main schemes. One of them is denoted as *Tikhonov regularization*, which is the scheme of interest for this project due to its wide application in different areas [Hansen, 1998]. It takes the form

$$\min\left\{\|Gm - d\|_2^2 + \lambda^2\|m\|_2^2\right\} \tag{B.24}$$

Where λ is a specified weighting factor. The objective of the proposed modified problem presented in Equation B.24 is that a regularized solution which residual is relatively small and simultaneously satisfies the constraint multiplied by λ becomes an stable solution of the underlying unperturbed problem [Hansen, 1998]. From statistics, the addition of the regularization term adds bias to the solution while decreasing the solution's covariance [Doyle, 2004].

A weakness of these regularization methods is that the solution procedure does not indicate how much regularization should be used [Doyle, 2004]. One of the most convenient tools for the determination of the parameter λ in Equation B.24 is the *L-curve approach*. The L-curve plot (see Figure B.3) is a plot for all the values of the regularization parameters and it displays the compromise between the norm of the residual $\|Gm - d\|_2^2$ and the regularized solution semi-norm $\|m\|_2^2$. This graph (see Figure B.3) usually takes a characteristic shape of "L" that gives it the name. The value that is located at the corner of the L-curve represents the "optimal" amount of regularization that balances the error induced by both terms on Equation B.24 [Hansen, 1998].

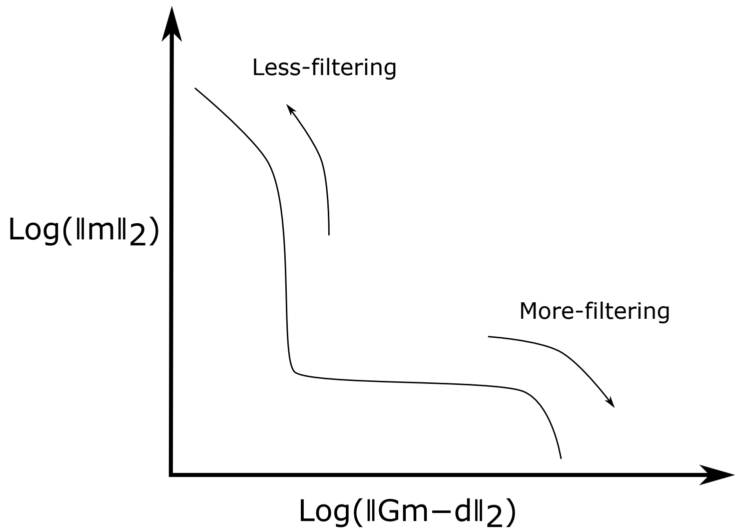


Figure B.3: L-curve form, note that the axes are in Log scale.

It is impossible to build any regularized solution below the curve, all the regularized solutions must lie onto the L-curve [Hansen, 1998].

If the regularized solution takes the form $m_{\text{reg}} = G^{\#}d_{\text{true}} + G^{\#}\eta$, where $G^{\#}$ denotes the regularized inverse. Thus, the error between the regularized solution and the original solution is given by

$$m_{\text{true}} - m_{\text{reg}} = \left(m_{\text{true}} - G^{\#}d_{\text{true}}\right) - G^{\#}\eta \tag{B.25}$$

When very little regularization is introduced, then the error between the regularized solution and the solution to the system is dominated by the noise η of the perturbed input data. This is denoted as *undersmoothing* and corresponds to the vertical part of the L-curve. On the contrary, when the amount of regularization is high, the error is dominated by the regularization error. This is called *oversmoothing* and corresponds to the horizontal part of the L-curve [Hansen, 1998].

C | Theory on Uncertainty Characterization

In this appendix, the methods used to quantify the uncertainty of the response of a system when the inputs are known, are presented. First, an overview of the concepts of error and uncertainty is given. Afterwards, the procedure for the characterization and propagation of uncertainties is presented, wherein type-A and type-B evaluations are defined, together with the coverage factor. Finally, the Monte Carlo (MC) method is introduced as a method to characterize uncertainties when analytic tools are not available.

When any experimental procedure is performed with the aim of obtaining a set of data which is to be used in a further analysis, it is of capital importance the correct assessment of the errors and uncertainties which are present in the whole process. In the case of an experiment in which the data obtained is accurate and trustworthy, it always must be accompanied with an associated uncertainty; otherwise, it is worthless [Mouritsen, 2013].

As measurements are repeated on apparently the same conditions (machines, specimens, environmental conditions, etc.), the values obtained i.e. the measurand, generally differ with respect to each other and the nominal value due to imperfections. The result of these imperfections is what is known as the *error* and it is identified from the following factors.

1. **Large errors:** Large errors are related directly to a poor execution of an experiment i.e. misuse of the equipment, misreading or faulty equipment among other reasons. These errors are unacceptable because they produce data that cannot be used. However, these are also the errors that can be identified more easily and are mostly avoided.
2. **Systematic deviations:** This group includes the one-sided deviations that give an unilateral shift to the result and are always present. They are directly linked with the measuring device and its precision; therefore inevitable and cannot be reduced. These errors are constant and range in between known values which can be estimated by repeating measurements.
3. **Random deviations:** It is a component of the error that varies unpredictably when measurements are being repeated even under the exact same conditions, but the obtained data fluctuates around a nominal value. It is impossible to tackle the random error of a measurement, but it can be reduced if the sample is increased. The estimate of this type of error is based on statistical methods, e.g. standard deviation or variance [Kirkup and Frenkel, 2006].

The fact of having errors in the measurement procedure precludes the possibility of knowing the real value of the measurand. The concept of *uncertainty* of a measurement is the statistical value which quantifies the variability of the measurand. The uncertainty characterizes the scatter of the values which could be associated with the measurand [JCGM-100, 2008].

Analysis of measurements using statistical methods is essential for the uncertainty characterization. The first step is to identify all the uncertainty sources, while afterwards they must be quantified. Some examples of uncertainty sources which can be found in an experiment are stated in the following list [JCGM-100, 2008].

- Definition of the measurand
- Choice of the measurement sample
- Inexact values of constants
- Test conditions (temperature, humidity, etc.)
- Measuring instrument and its usage

In the following sections, the different procedures to deal with uncertainty quantification are presented.

C.1 Propagation of Uncertainties

The current section presents an approach for evaluating the uncertainty associated to a measurand y. Normally, the magnitude of y is not determined from a direct measurement, i.e. it is an indirect measurement. It is determined from the contribution of other variables. These quantities and the measurand are related through a functional relationship [Kirkup and Frenkel, 2006] such as

$$y = f(x, z, ...) \tag{C.1}$$

In this particular section, it is assumed that the variables which conform the measurand are not correlated, therefore the further calculation of the uncertainties is simplified.

The uncertainty evaluation of the parameters which compound the measurand can be done by repetition of the measurements, resulting in a type-A evaluation. When it is impossible to perform a type-A evaluation then the type-B evaluation is chosen. The following paragraphs are based on the work presented in JCGM-100 [2008].

C.1.1 Type-A Evaluation

Type-A uncertainty evaluation relies on the analysis of uncertainty estimates from statistics, since n independent observations are obtained from the repeated measurements, i.e. a sample. The parameters of interest are the arithmetic mean $\bar{x}$, the experimental standard deviation $s(x)$, and the standard error (experimental standard deviation of the mean) $s(\bar{x})$. The expressions that define these parameters are stated in the following, where n is the number of observations.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{C.2}$$

$$s(x) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}))} \tag{C.3}$$

$$s(\bar{x}) = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \bar{x}))} \tag{C.4}$$

The value the standard error quantifies how far (or close) is the mean value of the sample from the population mean. Therefore, for a type-A evaluation of a magnitude X, the *type-A standard uncertainty* $u(x)$ of the measurand is the experimental standard deviation of the mean, such as

$$u(x) = s(\bar{x}) \tag{C.5}$$

C.1.2 Type-B Evaluation

This type of evaluation is only meant to be used when the type-A method is impossible to apply. This method relies on the scientific judgement using the type of data e.g. calibration documents, user experience, knowledge from literature or knowledge of the manufacturer of the machines. When it is only possible to estimate a lower/upper limit for the range of the magnitude (x^-, x^+), a statistical law must be chosen to obtain the standard deviation of the mean, and consequently, the *type-B standard uncertainty* $u(x)$. One of the theories proposed, which is considered to be the most robust is to assume that value can take any of the values inside the limits known [Rabinovich, 2005]. This is known as the *uniform law*. Equation (C.6) is the expression of the standard deviation i.e. the standard uncertainty of an uniform distribution.

$$u(x) = \frac{x^+ - x^-}{\sqrt{12}} \tag{C.6}$$

If it is considered that external values are less probable to appear then a trapezoidal or a triangular law can be adopted instead.

C.1.3 Combined Standard Uncertainty

As stated earlier in this chapter, the measurand y is obtained from the information of other input variables $(x_1, x_2, x_3, ...)$, which are not correlated. Therefore, the value of the standard uncertainty of y can be calculated by combining the standard uncertainties of the input values. Both type-A and type-B. The procedure of combining the uncertainties is also known as the law for accumulation of uncertainties [Mouritsen, 2013], see Equation (C.7).

$$u(y) = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} u(x_i) \right)^2} \tag{C.7}$$

The above expression is based on a first order Taylor approximation of the functional relationship f. The partial derivative $\partial f / \partial x_i$, is known as the sensitivity coefficient, and it quantifies how the measurand y changes with respect to the input value x_i.

There are two particular cases in the law for accumulation of uncertainties, when the functional relationship is a sum or a subtraction. In these cases, the accumulated uncertainty is obtained by adding the standard uncertainties of the inputs. The following expression exemplifies these cases.

$$y = x_1 \pm x_2 \quad \Rightarrow \quad u(y) = \sqrt{u(x_1)^2 + u(x_2)^2} \tag{C.8}$$

It can be the case where the level of confidence brought by the standard uncertainty calculated is not representative enough. Recall that the concept of standard uncertainty is directly linked to the experimental standard deviation of the mean. Therefore, it might be desired to enlarge the confidence interval of the measurand with the help of the *coverage factor k*. The uncertainty, is then multiplied by the coverage factor, thus obtaining the *expanded uncertainty*. In the case of the measurand being normally distributed, which frequently occurs in practice, it can be assumed that taking $k = 2$ will result in a confidence interval of approximately 95%. If $k = 3$, then the confidence interval is roughly 99% [JCGM-100, 2008].

C.2 Monte Carlo Methods

The approach described in Subsection C.1.3, the analytic combination of uncertainties, relies on a series of assumptions, e.g. the first order Taylor approximation of the functional, that makes it hardly applicable to problems resembling a highly nonlinear nature. Moreover, the analytical approach needs from an expression of a functional relationship. A common example are the nonlinear inverse problems, which usually resemble severe nonlinearities and no analytic expression of the functional is available. MC methods are a widely used alternative with proven effectiveness for cases where the analytic approach is not capable of representing the complexity of a system [Mosegaard and Tarantola, 1995].

MC methods are a set of computer-based algorithms employed to simulate "real-life" conditions of complex systems by means of repeated random sampling. When the process of interest is aimed to be represented by a deterministic numerical model, the aforementioned randomness is introduced "synthetically" in order to emulate the inherent unpredictability of "real-life" processes with a mere deterministic model. Relying on the Law of Large Numbers (see *"A Modern Introduction to Probability and Statistics (2005)"* for a detailed explanation of the Law of Large Numbers), the elemental idea of MC methods is repeating the random numerical sampling of the deterministic model a sufficient number of times to obtain a sample size which allows to estimate the quantities of interest of the population, e.g. mean value, with a reasonable accuracy [Kroese et al., 2014].

In experimental solid mechanics, it is a common practise to tune numerical models by modifying some parameters in order to fit experimental data, obtaining then by inverse parameter identification a desired property. MC simulations have been used for this purpose, where not only the expected value of the required model parameters is obtained, but also its confidence interval [Beaurepaire and Schüeller, 2011][Joubert and Marwala, 2016]. MC

Methods allow to estimate the sensitivity of the calculated output of an inverse parameter identification procedure to the variability of the input, and in doing so, the robustness of the inverse parameter identification routine is assessed. This is of great use in this project considering that an essential topic is obtaining a confidence interval of the results of the inverse parameter identification.

C.2.1 Theory of Correlation

The power of MC simulations relies on the generation of random numbers, which allows to examine processes for which any analytic approach is unviable [Kroese et al., 2014]. However, this randomness makes results more difficult to interpret as the link between input and output parameters is distorted by the randomization process. This problem can be overcome by correlation analyses, which provide a tool to analyze how much a change in a single input parameter influences the change of an output even if randomness is involved, as it is shown in Chapter 4. It calculates an estimate of the sensitivity of the results to changes in each of the inputs, which allows the analyst to identify the key parameters. In the case of the actual inverse parameter identification, the parameters which have a higher impact on the uncertainty of the result can be determined.

Correlation analyses are a branch of mathematics which deals with the statistical relationship between variables, i.e. how strong is the statistical association of two or more parameters [DeCoursey, 2003]. This is translated in a great tool for postprocessing MC simulation results, as it can provide an estimate of the influence of each of the input parameters onto the results.

Two measurements of correlation have been used in this project: *Pearson Product-Moment Correlation Coefficient* and *Distance Correlation*. Both have been used with the same purpose, but they offer different properties.

Pearson Product-Moment Correlation Coefficient (PPMCC), r_{xy}, measures how strong is the linear relation between two variables. For two variables x and y, PPMCC is defined for a set of values (x_i, y_i) as

$$r_{xy} = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\left[\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2\right]^{1/2}} \tag{C.9}$$

where $\overline{x}$ is the arithmetic mean of x_i and $\overline{y}$ is the arithmetic mean of y_i. It can be seen that PPMCC is dimensionless, which makes it more attractive as its value is independent of the units of the measurements [DeCoursey, 2003].

It can be shown that r_{xy} always falls in the interval $[-1, 1]$. If the pairs of values (x_i, y_i) form a straight line with positive slope, $r_{xy} = 1$. If they form a straight line with negative slope, $r_{xy} = -1$. If there is no linear relation between variables, $r_{xy} \approx 0$. Other values are an estimate of how close is the actual set of data to any of these situations [DeCoursey, 2003].

One way to understand PPMCC, as shown in Schmid [1947], is by comparing the results of performing two linear regressions: one considering x_i as the dependent variable, and other one considering x_i the independent variable (for an explanation of linear regressions analyses go to Section B.2 in Appendix B).

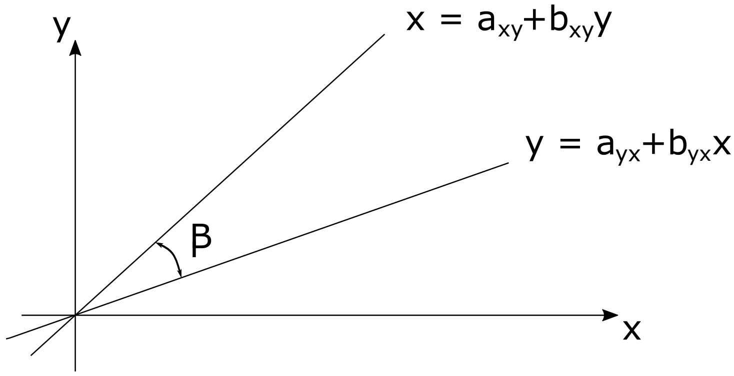


Figure C.1: Regression lines resulting from performing a regression analysis of the values (x_i, y_i) considering x_i as the observed variable and y_i as the observed data.

The resulting straight line equations are

$$x = a_{xy} + b_{xy}y \tag{C.10}$$

$$y = a_{yx} + b_{yx}x \tag{C.11}$$

where a_{xy} and b_{xy} are the vertical intercept and the slope, respectively, of the line when y_i are the observed values, and a_{yx} and b_{yx} are the vertical intercept and the slope, respectively, of the line when x_i are the observed values. It is easy to see that if the values (x_i, y_i) form a straight line, Equations (C.10) and (C.11) will represent the same line. However, as the data points (x_i, y_i) get further from a straight line, the lines in Equations (C.10) and (C.11) will differ more significantly. PPMCC is related to the cosine of the angle β in between these two lines, see Figure C.1. The exact relation is [Schmid, 1947]

$$r_{xy} = r_{yx} = \frac{1 - \sqrt{1 - \cos\beta}}{\cos\beta} \tag{C.12}$$

This corollary is not used in this project for the data analysis but for and understanding of the tool.

PPMCC is easily implemented and cheaply computed, which makes it a strong tool to find statistical association between variables [DeCoursey, 2003]. However, the assumption of linear relation is a great limitation. In the actual project, there is no clue on the relation between the results of MC simulations and input variables in this project. Thereby, the linearity of the response cannot be assured. For this reasons, a more complex, but more powerful measure of correlation is included, *Distance Correlation*.

Distance Correlation was developed by Székely et al. [2007] with the objective of creating a coefficient analogous to PPMCC but with a more general applicability. It is based on Euclidean distances between data points. It always lies on the interval $[0, 1]$, where 1 means total correlation and 0 no correlation at all. In contrast to PPMCC, *Distance Correlation* is 0 only and only if x and y are independent.

For a set of values (x_i, y_i), *Distance Correlation* is defined in the following. A series of matrices must be calculated in order to compute the value of *Distance Correlation*. For x_i

$$\begin{aligned} a_{kl} = |x_k - x_l|_p, \qquad \overline{a}_{k\cdot} = \tfrac{1}{n}\textstyle\sum_{l=1}^{n} a_{kl}, \qquad \overline{a}_{\cdot l} = \tfrac{1}{n}\textstyle\sum_{k=1}^{n} a_{kl} \\ \overline{a}_{\cdot\cdot} = \tfrac{1}{n}\textstyle\sum_{k=1}^{n}\sum_{l=1}^{n} a_{kl}, \qquad A_{kl} = a_{kl} - \overline{a}_{k\cdot} - \overline{a_{l\cdot}} + \overline{a}_{\cdot\cdot} \end{aligned} \tag{C.13}$$

for $k, l = 1 \ldots n$. And for y_i the analogous matrices are

$$\begin{aligned} b_{kl} = |y_k - y_l|_p, \quad \overline{b}_{k\cdot} = \tfrac{1}{n}\textstyle\sum_{l=1}^{n} b_{kl}, \quad \overline{b}_{\cdot l} = \tfrac{1}{n}\textstyle\sum_{k=1}^{n} b_{kl} \\ \overline{b}_{\cdot\cdot} = \tfrac{1}{n}\textstyle\sum_{k=1}^{n}\sum_{l=1}^{n} b_{kl}, \quad B_{kl} = b_{kl} - \overline{b}_{k\cdot} - \overline{b}_{l\cdot} + \overline{b}_{\cdot\cdot} \end{aligned} \tag{C.14}$$

for $k, l = 1 \ldots n$. The dots in the subindeces of a matrix indicate sumation in the direction of the index which is substituted by a dot, as it can be seen in Equations (C.13) and (C.14). A variable called *distance covariance* is defined from the matrices in Equations (C.13) and (C.14) as

$$\mathcal{V}_n^2(x, y) = \frac{1}{n^2} \sum_{k,l=1}^{n} A_{kl} B_{kl} \tag{C.15}$$

Similarly

$$\mathcal{V}_n^2(x) = \mathcal{V}_n^2(x, x) = \frac{1}{n^2} \sum_{k,l=1}^{n} A_{kl}^2, \qquad \mathcal{V}_n^2(y) = \mathcal{V}_n^2(y, y) = \frac{1}{n^2} \sum_{k,l=1}^{n} B_{kl}^2 \tag{C.16}$$

Finally, the empirical *Distance Correlation* $\mathcal{R}_n(x, y)$ is defined as

$$\mathcal{R}_n(x, y) = \begin{cases} \sqrt{\frac{\mathcal{V}_n^2(x,y)}{\sqrt{\mathcal{V}_n^2(x)\mathcal{V}_n^2(y)}}}, & \mathcal{V}_n^2(y)\mathcal{V}_n^2(x) > 0 \\ 0, & \mathcal{V}_n^2(y)\mathcal{V}_n^2(x) = 0 \end{cases} \tag{C.17}$$

This coefficient has a higher applicability than PPMCC, identifying nonlinear correlation between variables. Yet, it is a more computationally expensive measure due to the higher number of operations required to calculate it. It is decided to use both coefficients to be able to identify linear discern linear and nonlinear correlations.

It is worth mentioning that correlation does not necessarily imply dependence. As mentioned before, correlation analyses provide the statistical association between variables, which should not be understood as a true "cause-and-effect" relationship. The effect of having high correlation values with no true dependence between events is termed *spurious correlation* in the literature. This phenomenon appears in between two variables when there is an undiscovered third parameter influencing their value leading to an apparent direct relation in between them. When using correlation analyses, this effect should be present and the correlation results should be validated [Aldrich, 1995].

D Third semester project summary

In this appendix, the ideas behind the methodology developed during the third semester project Viejo et al. [2019] are explained to show the motivation of the work done in this project. First, the physical problem and a simple model are shown to demonstrate the main underlying concept which motivated the work done. Second, the model developed in the previous semester project is explained, as it is the core of the methodology developed in this project.

D.1 Double Cantilever Beam

The objective of this project is to design a delamination experiment to obtain the cohesive law of a material. For this purpose, a Double Cantilever Beam (DCB) experiment is chosen as the one shown in Figure D.1, where a force is applied at the end of the beams to delaminate the specimen. The reason behind choosing the setup is because it has been used before in the Department of Materials and Production at Aalborg University.

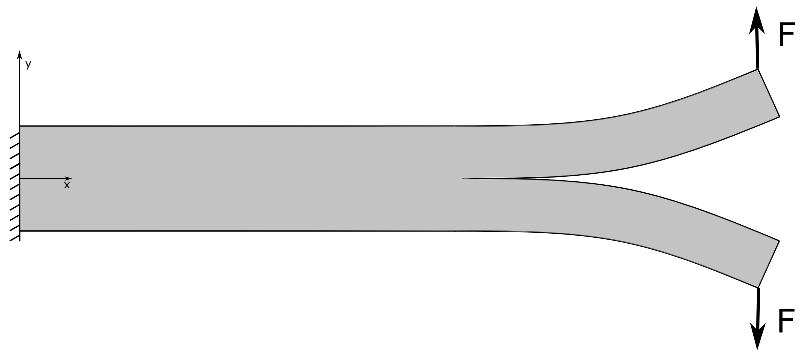


Figure D.1: Double Cantilever Beam with forces applied to provoke the delamination.

In Appendix A, it is explained that the cohesive law is the relation between the separation of the two surfaces resulting from the crack propagation at the interface and the traction which opposes this opening. The interface separation is easy to obtain as it is a relative displacement which can be measured directly on the faces of the specimen. The Digital Image Correlation (DIC) technique is used for this purpose. However, the tractions cannot be measured experimentally. For this purpose, it is decided to create model which resembles the DCB experiment which, for a given traction field, provides the separation of the interface. As explained in the Chapter 1, the tractions are going to be obtained by an inverse parameter identification procedure in which the traction field obtained is the one which minimizes the discrepancies between this numerical model and the real experiment.

Building a model which can simulate the cohesive zone behaviour introduces two main difficulties: the constitutive relation at the interface is nonlinear and the phenomenon is localized. This has been overcome in many studies by complex numerical models embedded in nonlinear optimizers, resulting in long computations and convergence issues [Shen et al., 2010][Manshadi et al., 2014][Jensen et al., 2019]. In order to avoid this, it is decided to simplify the model by removing the cohesive zone and substituting it for the tractions at the interface which appear during the delamination process. This process is sketched in Figure D.2, where the DCB specimen is transformed into a single beam with the traction field $T(x)$ which bonds the two beams together. The resultant configuration and the original one have an identical physical behaviour whenever the applied tractions $T(x)$ are the same as the traction field at the interface in the original problem.

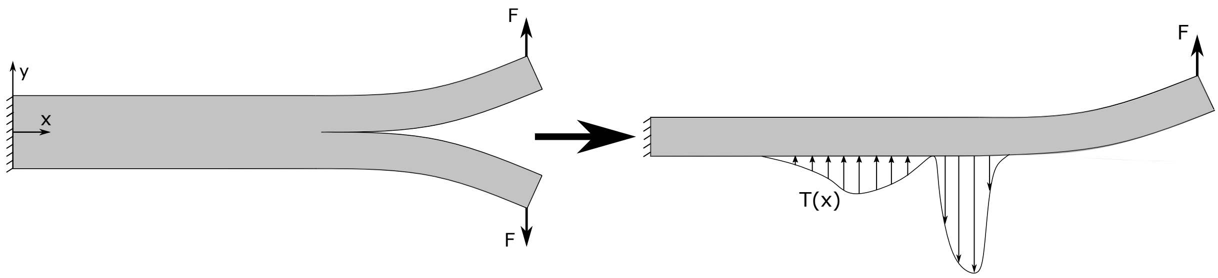


Figure D.2: Sketch of the beam simplification done to the DCB configuration.

Since the rest of the material is assumed to behave linearly, the whole problem becomes linear after removing the cohesive zone. Accordingly, a linear regression can be used to calculate the tractions, which ensures a faster and easier calculation with a guaranteed global minimum. See Section B.2 for more information.

D.2 Beam-based Model

The concepts behind the methodology are the same as the ones developed in Viejo et al. [2019]. Therefore, the information in this section is extracted from this third semester project.

To avoid any misunderstanding, all numerical models of a beam used to model the DCB experiment, Beam-Based models, are going to be termed BB-model. Doing this, it is easier to differentiate these numerical models from other models along this report, e.g. from Finite Element models.

D.2.1 Simple Model

To further simplify the problem, the pieces in which the DCB specimen is separated after the delamination are modelled as beams. Moreover, taking advantage of the symmetry of the mode I delamination problem, only one beam is modelled. The boundary conditions are the force applied for the delamination on one side and a clamp on the other side. Although in the real experiment there is not a real clamp but a free end, the behaviour is the same. This is because the symmetry of the DCB configuration ensures no reaction force nor moment at the end of the specimen in the vertical direction and there is no external force applied in the horizontal direction.

In order to show how the tractions are calculated from the displacement field, a simple example in which vertical point forces P_j are applied at different locations l_j of a beam is presented in Figure D.3. The displacement of a given configuration is simulated making use of beam theory. After that, the values of the forces are calculated using the previously emulated response, in an inverse parameter procedure.

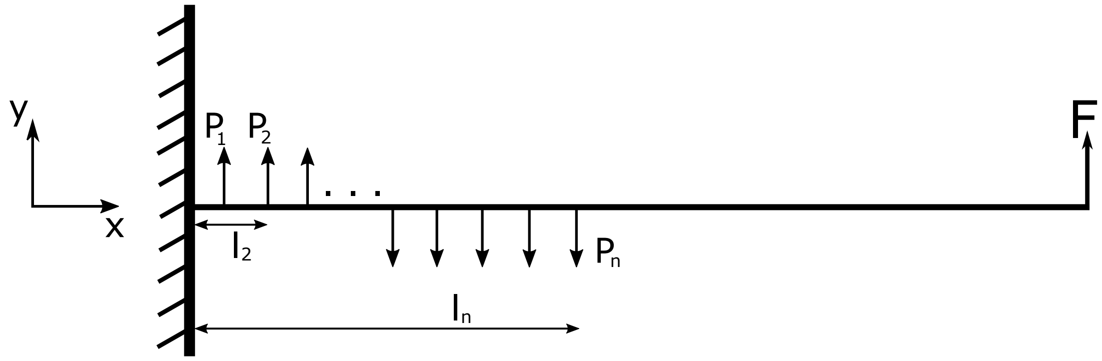


Figure D.3: Cantilever Beam model with point forces at different locations though the beam and a force applied at the end.

In this problem, the displacement values v_i at different points of the beam x_i are known, because they have been calculated previously to emulate an experimental response. The position of the point forces l_j, the geometry, and the material properties are also known parameters. Due to the linearity of the BB-model, the final displacement can be expressed as the sum of the contributions of each point force P_j and the force applied at the right hand side F. The beam displacement due to a single point force P at a given location l according to Bernoulli-Euler Theory follows Equation (D.1).

$$v_f = Pf(x,l) = \begin{cases} P\frac{x^2(3l-x)}{6EI} & \textit{for} \quad 0 \leqslant x < l \\ \\ P\frac{l^2(3x-l)}{6EI} & \textit{for} \quad l \leqslant x \leqslant L \end{cases} \tag{D.1}$$

where E is the Young's Modulus, I is the moment of inertia, L is the total length of the beam and x is the longitudinal coordinate of the beam which has a zero value at the clamp and L at the other end. f is the compliance function which only depends on the geometry, material properties and force position, so it is known for this problem. Finally, the total displacement v_i at a given location x_i is written as the sum of the displacement produced by each force P_j.

$$v_i = P_1 f(l_1, x_i) + P_2 f(l_2, x_i) + \cdots + P_n f(l_n, x_i) + Ff(L, x_i) \tag{D.2}$$

If there are displacement values v_i at different locations x_i the Equation (D.2) can be expressed in matrix form, resulting

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} f(l_1,x_1) & f(l_2,x_1) & \cdots & f(l_n,x_1) & f(L,x_1) \\ f(l_1,x_2) & f(l_2,x_2) & \cdots & f(l_n,x_2) & f(L,x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(l_1,x_m) & f(l_2,x_m) & \cdots & f(l_n,x_m) & f(L,x_m) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \\ F \end{bmatrix} \tag{D.3}$$

If the force F is known it goes to the left hand side

$$\begin{bmatrix} v_1 - Ff(L,x_1) \\ v_2 - Ff(L,x_2) \\ \vdots \\ v_m - Ff(L,x_m) \end{bmatrix} = \begin{bmatrix} f(l_1,x_1) & f(l_2,x_1) & \cdots & f(l_n,x_1) \\ f(l_1,x_2) & f(l_2,x_2) & \cdots & f(l_n,x_2) \\ \vdots & \vdots & \vdots & \vdots \\ f(l_1,x_m) & f(l_2,x_m) & \cdots & f(l_n,x_m) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \tag{D.4}$$

The inverse parameter identification can be done with Equations (D.3) or (D.4) where the unknowns P_j can be isolated as a function of the problem data, E, I, l_j and x_i, and a given displacement response v_i. Using the same notation as in Appendix B, Equations (D.3) and (D.4) are written

$$[G]\{m\} = \{d\} \tag{D.5}$$

It is easy to see that for having a closed system of linear algebraic equations, with a single solution of P_j, the same amount of displacement data points v_i are needed as there are unknowns P_j. However, it only works if the displacement data corresponds exactly to the actual BB-model, otherwise the system is probably inconsistent. Meaning that no combination of point forces would match exactly the given displacements and the algebraic system of equations has no solution, as explained in Section B.1. Real systems have discrepancies between physical models and experimental data and they are always subjected to noise. Consequently, it is common to minimize the discrepancies between the model and the experiment instead of seeking an exact match, which is done by making use of the Equation (B.6).

D.2.2 Advanced Model

The main workload of Viejo et al. [2019], is the development of a BB-model based on the methodology explained in the above paragraphs, seeking a higher accuracy in the obtained cohesive law. The main modifications done in Viejo et al. [2019] to the model to improve its accuracy are summarized in the following. All the calculations and numerical modelling in relation to the Beam-Based approach are done in the program MATLAB.

For the development of most features of the inverse parameter identification program a 2D Finite Element model (FE-model) done in ANSYS Mechanical was used to simulate an experimental response of the displacements. Thereby, the quality of the obtained tractions by the BB-model can be judged by comparing it with the traction field in the FE-model.

Distributed loads

The BB-model explained above predicts point forces, however, the tractions at the delamination interface are a continuous 2D field of stresses. Therefore, it is decided to derive another equation to substitute Equation (D.1) in order to make the displacement depend on continuous tractions instead of point forces. The concept explained in Section D.2.1 can only be used to calculate the magnitude of a load. The location l_i of this loads has a nonlinear dependency with respect to the displacements. Accordingly, the shape and position of the distributed load have to be set beforehand for the linear regression routine to work. The choice made is a triangular load $q(x)$ with a base ranging from a to c, where the traction value is zero, and a maximum value at b, shown in Figure D.4, for a beam of length L. It is important to emphasize that the triangle is not necessarily isosceles, a, b and c can have any value which fulfils $0 < a < b < c < L$.

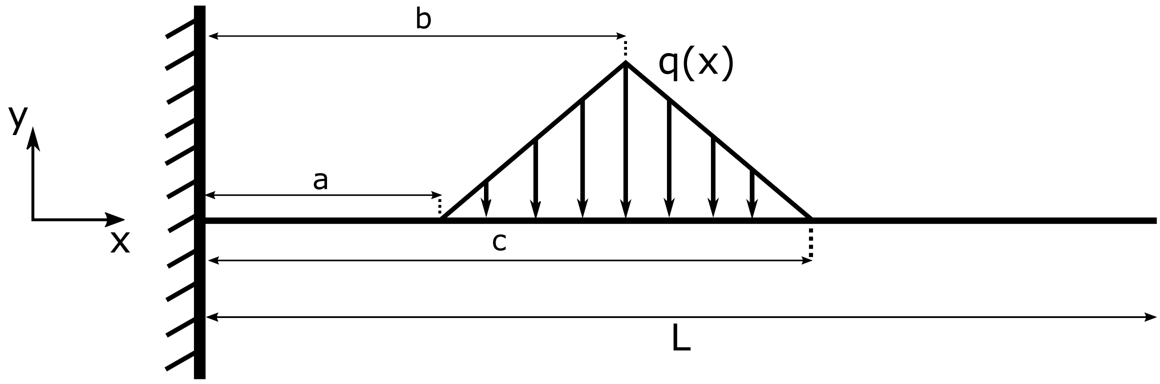


Figure D.4: Cantilever beam with a triangular load applied [Viejo et al., 2019].

The expression of the displacement v is derived using Euler-Bernoulli beam theory. The obtained equation is

$$v = qf(a,b,c,x) = \begin{cases} \dfrac{qx^2\,(a-c)(a+b+c-x)}{12EI} & \text{if } \; x < a \\[2ex] \dfrac{qx^2(a-b)(a+2b-x)}{12EI} - \dfrac{q(a-x)^5}{120EI(a-b)} + \dfrac{qx^2(b-c)(2b+c-x)}{12EI} & \text{if } \; a < x < b \\[2ex] \dfrac{q(-4b^5+5b^4c+15b^4x-20b^3cx+10c^3x^2-10c^2x^3+5cx^4-x^5)}{120EI(b-c)} \\ \quad -\dfrac{q(a-b)\left(a^3+2a^2b-5xa^2+3ab^2-10xab+4b^3-15xb^2\right)}{120EI} & \text{if } \; b < x < c \\[2ex] -\dfrac{q(a-b)\left(a^3+2a^2b-5xa^2+3ab^2-10xab+4b^3-15xb^2\right)}{120EI} \\ \quad -\dfrac{q(b-c)\left(4b^3+3b^2c-15xb^2+2bc^2-10xbc+c^3-5xc^2\right)}{120EI} & \text{if } \; c < x \end{cases} \tag{D.6}$$

where x is the longitudinal coordinate of the beam, E is the Young's modulus in the x direction, I is the moment of inertia, f is the compliance function and q is the value of the distributed traction peak, the value at b. In this case, the objective of the inverse parameter identification is the calculation of the value of q.

Again, making use of the linearity of the problem, the final displacement can be calculated as a combination of different linear cases. The position of this load has to be defined prior the calculation. Inspired by the shape functions of the Finite Element Method, the triangular loads are concatenated by setting the ends of a load where the peak of the closest ones are positioned, see Figure D.5. Which for a given load $q_i(x)$, this condition is expressed mathematically in Equation (D.7).

$$a_{i+1} = b_i = c_{i-1} \tag{D.7}$$

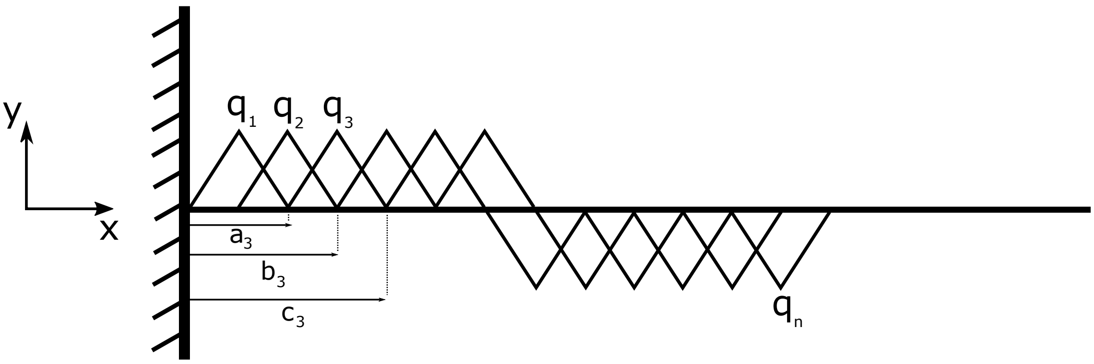


Figure D.5: Cantilever beam with evenly distributed triangular loads [Viejo et al., 2019].

After adding together all the triangular loads calculated by the inverse parameter identification, the resulting distribution is a piece-wise linear function with zero value at both extremes and changes in slope at the traction peaks . The stress field at the interface has a complex shape with high gradients. Therefore, the use of linear loads may lead to a high number of degrees of freedom if a faithful representation is required, leading to instability problems, i.e. an ill-conditioned system, see Section B.3. Anyhow, this configuration allows for flexible analysis and an intuitive use of the inverse parameter identification tool, simplifying the further development of the model and the understanding of the linear regression analysis behaviour.

Nonlinear optimization of the load positions

After using the displacement data of the FE-model on the built inverse parameter identification routine, it turns out that the accuracy of the obtained tractions is very dependent on the position of the calculated distributed loads (a_i, b_i, c_i). They have to be specified prior the linear regression as it is used to construct the matrix $[G]$. To overcome this, the linear inverse parameter program is embedded into a nonlinear optimizer which finds the optimum position of the traction peaks:

- The objective function is chosen to be the same one as the in the linear regression: the Euclidean norm of the residual of the displacements. Recalling Equation (B.4), the objective function r is

$$r = \|[G]\{m\} - \{d\}\|_2^2 = \sum_{i=1}^{w} \left(G_i^T m - d_i \right)^2 \tag{D.8}$$

where w is the number of displacement data points.

- The design variables are the locations of the triangular loads. The number of variables is reduced due to how the position of each triangle is linked to its neighbours, see Equation (D.7) and Figure D.5. Each peak is located at the same position as the two low vertices of its neighbours, thus, the number of variables is the number of peaks b_i and the two low vertices of the first and last triangles a_1 and c_n. n is the number of triangular loads. It is organized in one vector $\mathbf{x}$ in Equation (D.9).

$$\mathbf{x} = \begin{pmatrix} a_1 & b_1 & b_2 & \dots & b_j & \dots & b_{n-1} & b_n & c_n \end{pmatrix}^T \tag{D.9}$$

Some constraints are introduced in the optimization problem to avoid nonphysical results and numerical issues.

- The positions of the three vertices of the triangular load must remain in the same order.

$$a_i < b_i < c_i, \qquad \forall i \in [1, n] \tag{D.10}$$

- The tractions must be in the beam domain.

$$a_i, b_i, c_i \in [0, L], \qquad \forall i \in [1, n] \tag{D.11}$$

The optimization problem is is written in standard way as

$$\begin{aligned} &\underset{(\mathbf{x})}{\text{Minimize:}} \quad r = \| [G(\mathbf{x})] \left([G(\mathbf{x})]^T [G(\mathbf{x})] \right)^{-1} [G(\mathbf{x})]^T \{d\} - \{d\} \|_2^2 \\ &\text{Subject to:} \quad A\mathbf{x} \leq B \end{aligned} \tag{D.12}$$

A and B are the matrices which contain the linear inequality restrictions explained above. For seven design variables A and B become

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} , \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ L \end{pmatrix} \tag{D.13}$$

The impact of implementing a nonlinear optimization is that the solution of the problem is no longer analytic and the calculation time is longer, specially if the number of design variables is increased significantly. Nonetheless, this approach provides more accurate results with a lower number of degrees of freedom and the calculation time of a single traction field is still done in a short time.

The nonlinear optimization problem is solved using the MATLAB built-in function *fmincon* [MathWorks, 2019].

Timoshenko beam theory

The inverse parameter identification procedure was proven to be effective as it minimized the discrepancies between the displacement response of the FE-model and the beam deflection. Nonetheless, the obtained traction field differed considerably with respect to the FE-model. After studying the behaviour of the BB-model when applying the stress field from the FE-model, it turned out that the shear effects are of great importance in this problem. These strong shear effects are caused by the abrupt change in tractions at the interface were the cohesive law is active. To account for the shear effects it was decided to change to Timoshenko beam theory. A new expression for the compliance f is derived as

$$f = \begin{cases} \frac{x(a-c)\left(6EI-kAGx^2+AGakx+AGbkx+AGckx\right)}{12EIkAG} & \text{if } x < a \\[2ex] \frac{-kAGa^5+5kAGa^4x+20EIa^3-10kAGab^2x^2-10kAGabcx^2+10kAGabx^3-60EIabx-10kAGac^2x^2+10kAGacx^3}{120EIkAG(a-b)} \\ \quad + \frac{-60EIacx-5kAGax^4+60EIax^2+10kAGb^2cx^2+10kAGbc^2x^2-10kAGbcx^3+60EIbcx+kAGx^5-20EIx^3}{120EIkAG(a-b)} & \text{if } a < x < b \\[2ex] \frac{a^2+ab-2b^2}{6kAG} - \frac{(a-b)\left(a^3+2a^2b-5xa^2+3ab^2-10xab+4b^3-15xb^2\right)}{120EI} \\ \quad + \frac{-4b^5+5b^4c+15b^4x-20b^3cx+10c^3x^2-10c^2x^3+5cx^4-x^5}{120EI(b-c)} + \frac{2b^3-3b^2c+3c^2x-3cx^2+x^3}{6kAG(b-c)} & \text{if } b < x < c \\[2ex] \frac{(a-b)(20EIa+40EIb)}{120EIkAG} - \frac{(b-c)\left(4AGb^3+3AGb^2c-15AGxb^2+2AGbc^2-10AGxbc+AGc^3-5AGxc^2\right)}{120AEGI} \\ \quad - \frac{(a-b)\left(AGa^3+2AGa^2b-5AGxa^2+3AGab^2-10AGxab+4AGb^3-15AGxb^2\right)}{120AEGI} + \frac{(40EIb+20EIc)(b-c)}{120EIkAG} & \text{if } c < x \end{cases} \tag{D.14}$$

where G is the shear modulus, A is the area of the cross-section and k is the shear correction factor.

Performance against FE-data

These three changes in formulation allowed for a more accurate analysis, meaning that the obtained cohesive law was a close representation of the one introduced in the FE-model. The obtained traction field using the displacements of the FE-model is shown in Figure D.6 together with the extracted cohesive law.

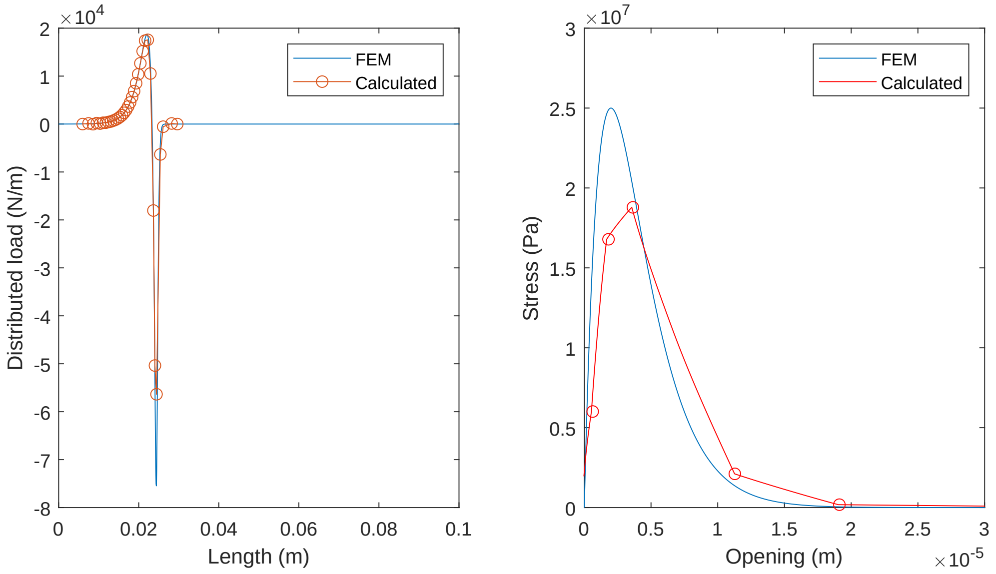


Figure D.6: Results obtained in Viejo et al. [2019] using the so-called "advanced" BB-model to identify the cohesive law from a FE-model displacement response. On the left, the identified tractions per unit width of the beam compared with the tractions of the FE-model. On the right, the obtained cohesive law and the cohesive law used in the FE-model.

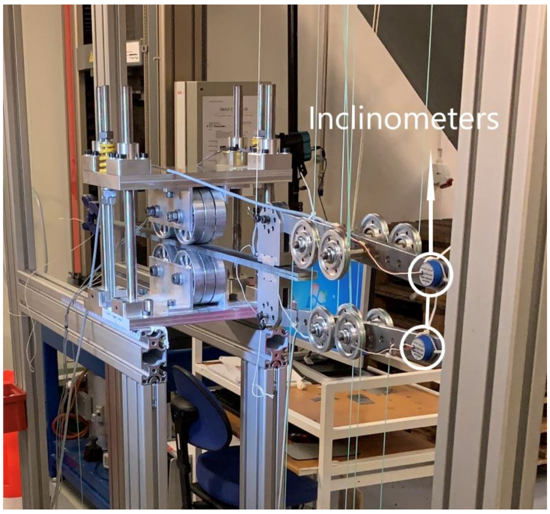
It can be seen that the program works as intended with some inaccuracies, especially the prediction of the traction peak. Nevertheless, the slope of this part is the most difficult to predict by another methods as well [Bak, 2017]. The more loads, the higher resolution of the traction field. However, for a higher number of loads than the one shown in Figure D.6 the solution becomes unstable providing nonphysical results. This behaviour is caused by the ill-conditioning of the system, as explained in Section B.3, setting the main limit for the accuracy of the calculation.

D.3 Original Experimental Procedure

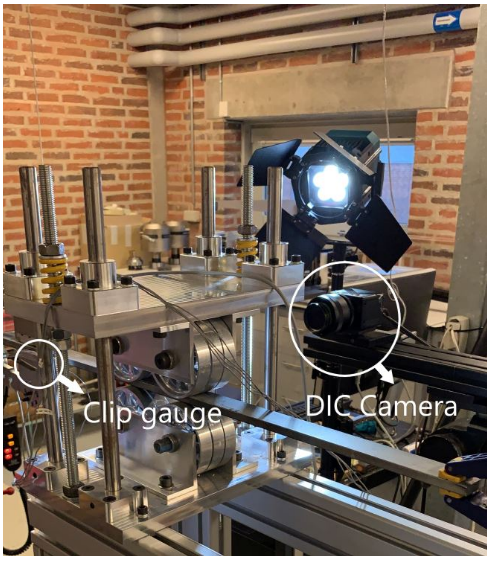
In the previous part of this appendix, the underlying concepts which inspired the project together with the development of the BB-model which was first presented in Viejo et al. [2019] are shown. In the next sections, the experimental setup and the procedure followed to acquire the required data to introduce in the BB-model are explained. The result is taken as the starting point of the actual project and the conclusions as motivation for the work preformed.

D.3.1 Third Semester Experimental Setup

During the course of the 3rd semester, a delamination experiment was carried out. However, in contrast with the experimental procedure which has been presented in Chapter 1 and Appendix D, the delamination process was moment-driven. The circumstance that even moments are applied at the tips of both beams causes the self-similarity of the crack. The fact of the crack being self-similar establishes a constant length of the FPZ, and a constant shape for the interface tractions once the crack is fully developed, i.e. from only one specimen, the number of cohesive laws which can be extracted is the same as the number of images of the developed crack that have been taken. As well as it has been stated in Chapter 1, the main objective of these experiments was the collection of the displacement field at the exterior faces of the DCB, and using this information as an input for the BB-model in order to obtain the interface traction field. The setup which was employed can be seen in Figure D.7, in which the different devices are highlighted. Figure D.7a) shows the specimen from one of the sides with the inclinometers placed in the arms, while Figure D.7(b) shows the other side of the setup with the DIC camera and the clip gauge.



(a)



(b)

Figure D.7: Experimental setup from both sides of the specimen [Viejo et al., 2019].

The device used to generate the moments onto the DCB is the "Smart" testing machine, which is an in-house instrument that is attached to a tensile test machine. This configuration was used due its the current accessibility and because of the simplicity of the assembly of the experiment. Through a combination of wires and rollers, the force delivered by the tensile machine is converted to a pair of moments; the ones which are applied to the specimen. What is of interest are the moment values, and the recorded data is the force applied by the tensile machine. However, if the rotation of the hinges is known, the value of the moment can be calculated. Therefore, a pair of inclinometers were placed wherein the moment is applied.

In order to obtain the displacement field of the arms, DIC was used. A high-resolution camera was placed with the lens parallel to the lateral face of the DCB. The distance between the camera and the specimen was set in order to capture the largest area possible and maintain the accuracy of the images. As the delamination process goes on, the software which drives the camera is set to capture one picture per second. Then, the displacement field is obtained from the postprocessing of these images.

The setup also included a clip gauge, which was attached to the specimen with the help of two metal rods, that were placed at the crack tip. The data that the clip gauge records is the vertical displacement of the crack tip i.e. the CTOD. With this data, it was desired to verify the results obtained from the DIC, as a validation procedure to ensure that the values which were being collected were good.

Previously mentioned, the desired outcome of the experiment was the collection of the moment and displacement data. However, three different software programs were used to register all the information. This fact enforced the temporal synchronization of all the devices, since a time lag would mean that the obtained data is not valid for the postprocessing. Furthermore, the faced experiment is a destructive test; only one attempt per specimen is possible.

D.3.2 Data Postprocessing and Results

The previously described setup was used to acquire the data of three specimens. Due to time limitations and problems in the performance of the experiment, only the data of one of the specimens was actually processed and used for the inverse parameter identification routine. The raw data obtained from the experiment is stated in the following.

- Load values applied by the tensile test machine along the whole delamination experiment.
- The angle of inclination of the arms used to apply the moment.
- Opening of the initial crack tip measured by the clip gauge.
- Pictures of the undeformed and deformed specimen at different stages of the delamination process captured by the camera.

The data needed to perform the inverse parameter identification of the traction field at the interface is the moment, applied by the arms, and the displacement of the centerline of the DCB beams. The moment is calculated with a simple formula which includes the value of the tensile test machine force, the inclination angle of the arms and geometric values accounting for the configuration of the wires and rollers used to convert the tensile force into moment.

The DIC analysis is performed using the software GOM correlate 2019. A study is conducted on the parameters of the analysis to ensure a valid result. In addition, the obtained displacement values are compared against the clip gauge validating them.

After further analyzing the data, it turned out that the values of the tensile test machine were not properly synchronized with the other measuring devices, resulting in wrong values of the moment applied. Furthermore, the opening rate of the tensile test machine was set too high, resulting in most of the pictures taken being perceptibly blurry, lowering considerably the quality of the displacement results. Many pictures were also discarded because the Fracture Process Zone lied outside the range of the camera.

Anyhow, the moment value was approximated by the curvature of the traction-free part of the beam and the inverse parameter identification was performed on the two pictures with best quality. Traction fields are displayed in Figure D.8 and the cohesive laws in Figure D.9.

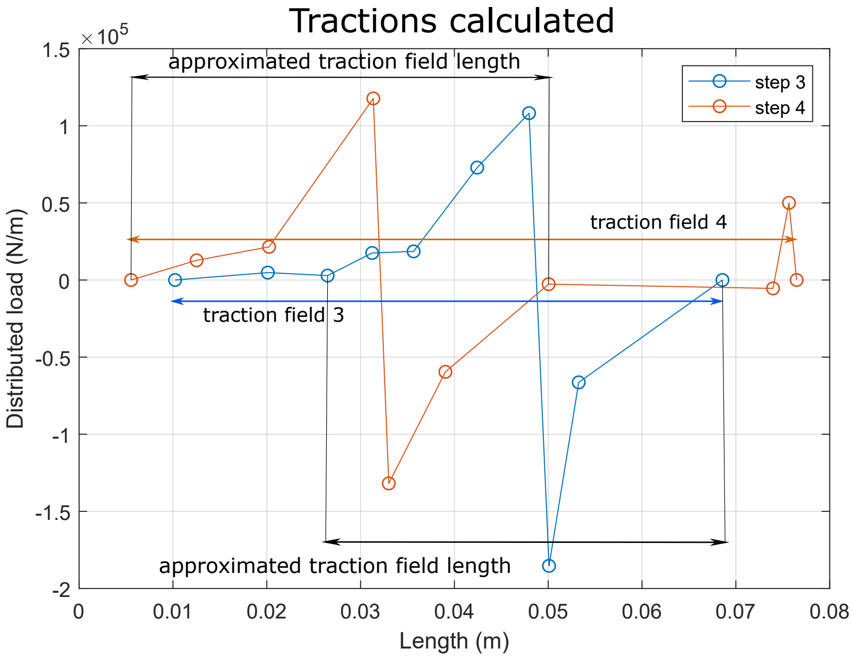


Figure D.8: Obtained traction fields for the two pictures analyzed [Viejo et al., 2019].

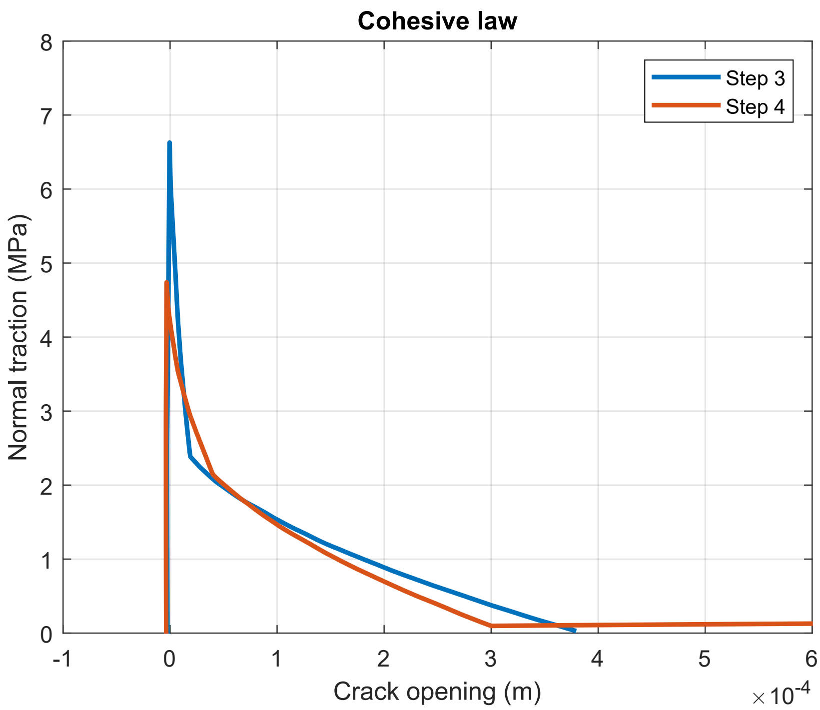


Figure D.9: Obtained cohesive laws for the two pictures analyzed [Viejo et al., 2019].

Conclusions and Present Work

Although several problems were encountered during and after the delamination experiment regarding the data acquired, the results obtained seemed physically possible. This conclusion is made from two observations:

- The crack is self-similar due to the pure moments applied. This is captured in Figure D.8, where the traction fields resemble each other even though they are calculated at different positions of the DCB.
- The cohesive laws in Figure D.9 are also similar, supporting the idea of being a material constant. However, the onset traction value differs considerably. This weakness of the method was known, as it has been shown in Subsection D.2.2.

After this results, it can be concluded that the method seems capable of achieving a good result as it can identify tractions of a FE-model closely and give feasible results for real data. However, no trust can be put in this results as the robustness of the methodology is unknown, the method has only been validated against deterministic data of a simplified 2D FE-model. There is little or no knowledge of the impact of all the parameters of the experiment on the obtained result. Therefore, assessing the performance of the procedure followed to obtain the cohesive law in terms of accuracy and robustness, and using it to develop a valid methodology for cohesive laws characterization is the objective of the present work. The evaluation is going to be performed by, first, characterizing the sources of error, and then, measuring its impact on the results. The work done is displayed as

- The data used to build the BB-model, i.e. material and geometric data, is characterized in terms of uncertainty in Chapter 3.
- The possible mismatch between the beam model and the full DCB problem are studied in Chapter 2 with the aid of a high-fidelity 3D FE-model.
- The impact of the sources of error on the inverse parameter identification routine is evaluated in Chapter 4. In addition, this data is used to develop the present methodology to achieve a satisfactory result in Chapter 5.

E | Building the FE-model of the Delamination Experiment

A FE-model is built using ANSYS Workbench. A CAD model of the DCB and the hinges is made based on the real components as shown in Figure 2.3. However, since forces are always applied vertically by the arms of the tensile machine, just the part of the hinge that is bonded to the beam is modelled. Thanks to this simplification, the number of connections to be defined and number of elements to be used makes the FE-model to use less resources and be faster in the computations.

E.1 Modelling Features

E.1.1 Material Properties

The materials used for the FE simulation are:

- Unidirectional (UD) fiber glass composite: DCB specimen
- Structural steel: hinges
- User-defined interface material: Delamination interface

The properties of the UD composite cannot be shown in this report due to privacy reasons. It has been proven, with different models, how the change of material of the hinges (as long as it is 1 order of magnitude stiffer than the DCB material) has a minimum impact of the final results obtained. Therefore, the default ANSYS Structural Steel is used for the hinges. Finally, an interface material must be defined for the delamination behaviour of the specimen, characterized by a user-defined cohesive law. In this particular case, it is previously known by external sources that the specimens used have a higher bridging compared to the ones used in the previous semester project. It is known from CZM that fiber bridging is represented by the last part of the cohesive law. Therefore an exponential cohesive law is modelled accordingly, with the same critical energy release rate (area under the cohesive law) as the one used in the third semester project, but a higher critical displacement to simulate bridging (see Figure E.1).

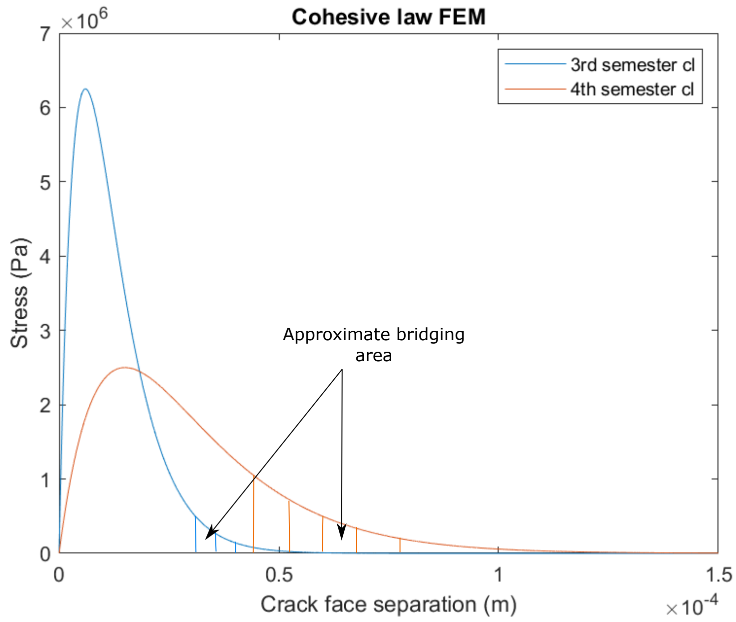


Figure E.1: Cohesive law data used in the FE software for the simulation.

E.1.2 Meshing

For the meshing of the model four areas/components are distinguished:

- Hinges
- Double cantilever beam (DCB)
- Delamination interface
- Contact surfaces

Hinges

For the hinges, there is no restriction of element characteristics, a simple patch conforming method with tetrahedral quadratic elements (SOLID187) is used (see Figure E.2). The maximum element size is set to 4 mm while the minimum element size is 2 mm. The sizing of the elements must be fine enough so the contact elements are able to transfer the loads properly in the connection face with the DCB specimen.

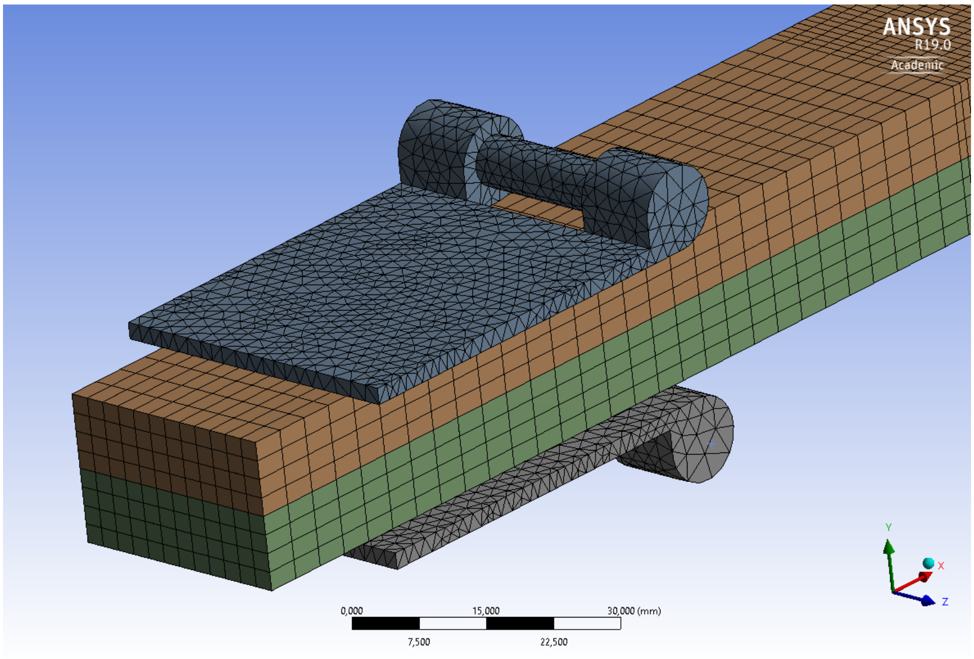


Figure E.2: View of the tetrahedral mesh used for the meshing of the hinges that transfer the load.

DCB

For the DCB specimen, a MultiZone method is chosen for a pure hexahedral mapped mesh creation. It is known from the Interface Delamination characteristics in ANSYS that for an implementation of this analysis a mapped, quadrilateral mesh with matching nodes at the interface is preferable, as hexahedral elements are the ones that perform the best [Cook et al., 2002]. Therefore a uniform mapped mesh with hexahedral quadratic elements (SOLID186) is done. Due to the extreme amount of elements that this particular mesh needs, multiple edge sizings (shown in Figure E.3) are set in order to concentrate all elements in the area of interest.

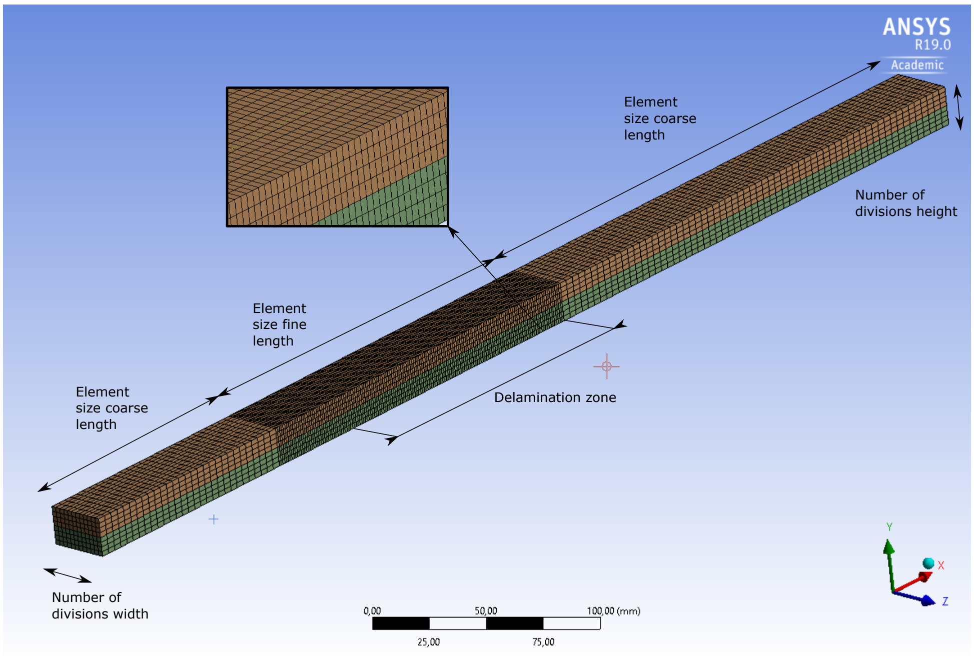


Figure E.3: View of the mesh used for the DCB specimen with the different sizings applied at each of the edges.

Delamination Interface

Due to the excessive amount of elements used when simulating delamination across the full interface of the DCB, it is chosen to model just the initial delamination area (Figure E.3). Apart from the element number issue, this simplification is also done due to the limited field of view that the camera has, which makes impossible to capture the delamination process along all the interface of the specimen.

3D quadratic interface elements (INTER204) are used for this analysis. The meshing process can be regarded as "indirect", as the elements are created from the solid elements used for the DCB mesh creation. For the creation of interface elements, the mesh at both surfaces of the interface must be equal i.e. nodes must be matching, and quadrilateral elements must be used.

Contact surfaces

For the link between hinges and the arms of the DCB specimen contact elements must be used. Even though the hinges in reality are bolted to the DCB specimen, it is chosen to be modelled as bonded, because the effect of the bolt holes are considered to be too far from the interest region of this study. Therefore, the type of contact used is general bonded contact, which automatically creates elements from the geometric features of the model, using CONTA174 and TARGE170 for meshing the contact surfaces between hinges and specimen.

E.1.3 Boundary Conditions

The boundary conditions for the DCB specimen must imitate, in a realistic way, the experimental conditions. In the case under consideration, the experiment is conducted in a tensile machine, where the arms of the hinges are pulled apart by the machine maintaining the specimen in equilibrium of forces. However, in the FE software, it is not possible to apply the boundary conditions shown in Figure E.4 in the top part, due to rigid body movement of the body. Therefore, the setup shown in the low part of Figure E.4 is chosen for a successful implementation of the experimental procedure in the FE software.

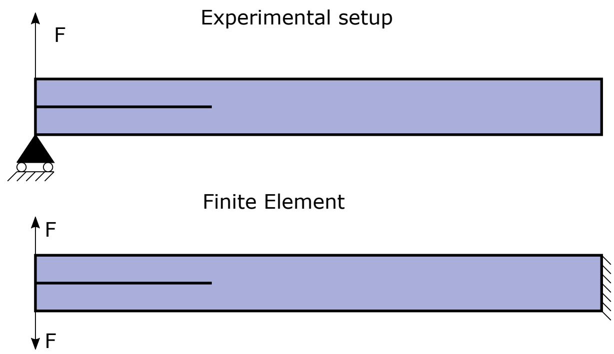


Figure E.4: Comparison between the boundary conditions applied for the analysis in the FE software and the real experimental setup.

The force constraint applied at the tip of the DCB is chosen to be displacement controlled, to ensure a stable crack growth and avoid snap through behaviour of the solution, assuring the convergence of the desired results. The option used is the remote displacement option, which makes use of an MPC contact to transfer the forces between a *remote point* and the geometry. The behaviour for this option is set to be deformable, to avoid excessive stiff behaviour of the FE-model, and the only movement constraint is in the Y direction, leaving free all the rest of DOF, both rotations and translations. The constraint is applied to the lower face of the hinge connection, as shown in Figure E.5.

For this analysis, as only a certain part of the DCB is being delaminated (see the delamination area in Figure E.3), the displacement needed for achieving delamination is of 1mm. It is applied in two steps as indicated in Figure E.6, the first step being from 0 mm to 0.5 mm and the second step from 0.5 mm to 1 mm.

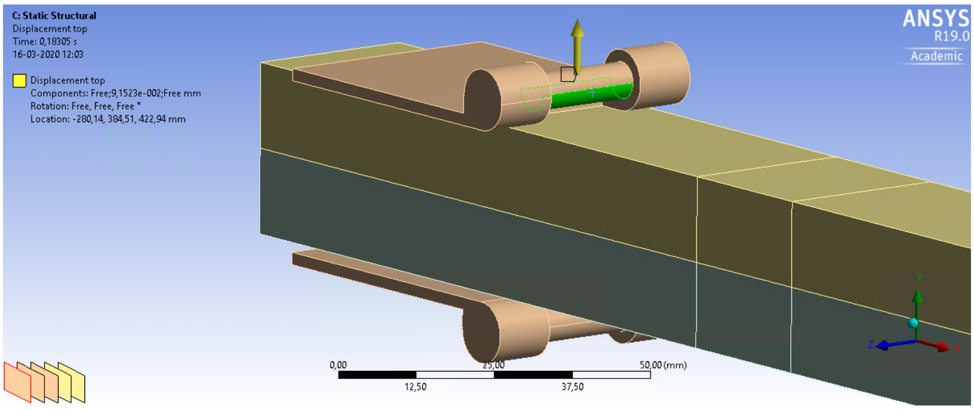


Figure E.5: Remote displacement applied at the highlighted (green) face of the hinge.

E.1.4 Solver Settings

As already mentioned, a delamination problem is a nonlinear problem. Therefore, the main focus for the solving procedure is to assure the convergence of the solution. The settings chosen for the convergence of the analysis are shown in Figure E.6.

Step Controls	
Number Of Steps	2,
Current Step Number	1,
Step End Time	1, s
Auto Time Stepping	On
Define By	Substeps
Initial Substeps	5,
Minimum Substeps	5,
Maximum Substeps	10,
Solver Controls	
Solver Type	Program Controlled
Weak Springs	Off
Solver Pivot Checking	Program Controlled
Large Deflection	On
Inertia Relief	Off
Rotordynamics Controls	
Restart Controls	
Fracture Controls	
Fracture	On
SIFS	Yes
J-Integral	Yes
Material Force	No
T-Stress	No
Nonlinear Controls	
Newton-Raphson Option	Program Controlled
Force Convergence	On
--Value	Calculated by solver
--Tolerance	0,5%
--Minimum Reference	1,e-002 N
Moment Convergence	Program Controlled
Displacement Convergence	Program Controlled
Rotation Convergence	Program Controlled
Line Search	Program Controlled
Stabilization	Off

Figure E.6: Settings used for the convergence of the analysis.

The solution for the analysis converges satisfactory as can be seen in the Force Convergence Plot (Figure E.7).

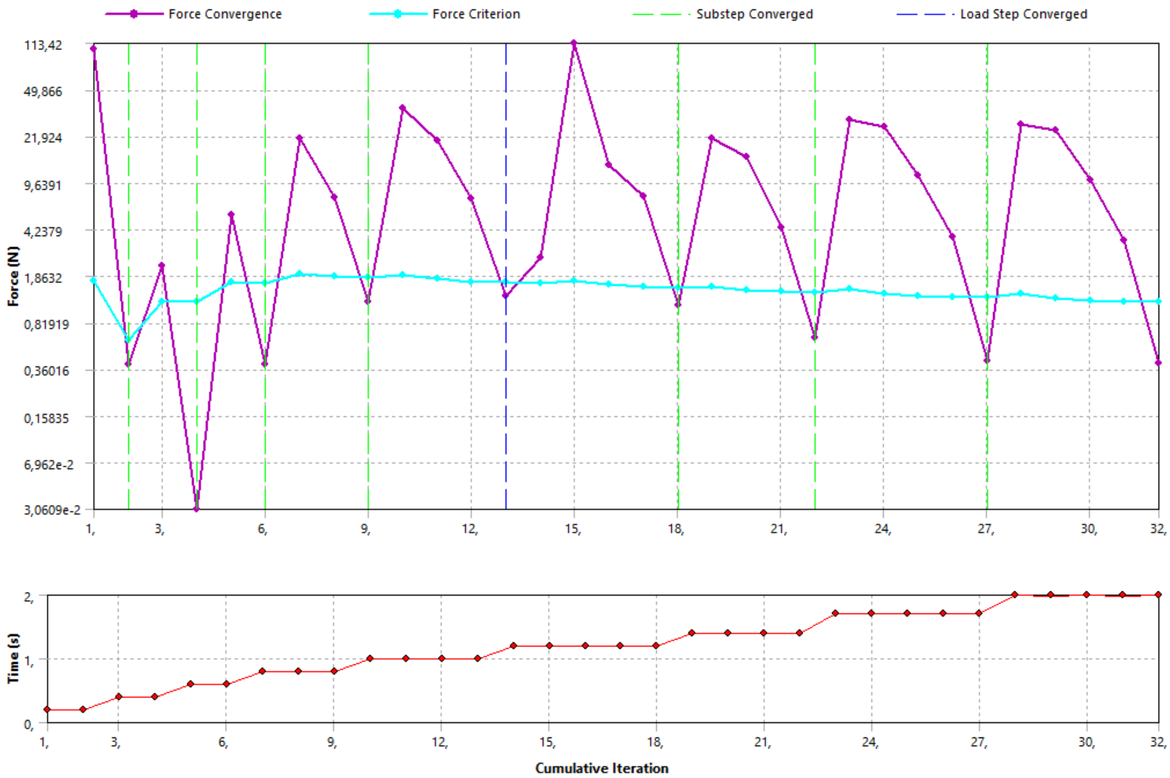


Figure E.7: Force Convergence Plot showing a converged solution with the previously mentioned analysis settings.

E.2 Validation of the Results

For a trustworthy validation of the results, a mesh convergence analysis must be performed. In this case, considering the non-uniform mesh manually created (see Subsection E.1.2) the refinement and study is also done manually. Due to the limited amount of nodes that the Student Version of ANSYS allows, elements must be used carefully and productively, for this reason the effect of the mesh refinement in all edges of the DCB is treated separately to evaluate the effect of the mesh on the results. For a nonlinear analysis, convergence of all the results simultaneously is hard to achieve, that is why only a few parameters are of interest for the mesh convergence study.

The results that are of more interest for the project are:

- Force reaction at the hinges
- Maximum and minimum normal stresses at the interface
- Maximum deformation along the width and maximum total deformation
- Force reaction at the fixture

The approach followed for the mesh convergence study is first of all to achieve a number of elements between 4-8 in the cohesive zone (as a rule of thumb the minimum number of elements needed for a correct representation of the cohesive zone is 4). Then, refinement along the length is prioritized, because it is known how this parameter is the most important to achieve a successful convergence [Viejo et al., 2019]. After convergence is achieved in one direction, the first solution that is considered converged is chosen for the refinement of the other parameters, until what is considered an accurate solution is achieved.

The settings for the mesh convergence study are presented in Table E.1.

Element size length coarse (mm)	**Element size lentgh fine (mm)**	**Number of divisions width**	**Number of divisions height**	**Number of nodes**	**Number of elements**
Refinement in length (x-direction)					
10	10	4	4	42180	20742
10	8	4	4	43080	20902
10	6	4	4	44486	21142
10	4	4	4	47220	21638
10	2	4	4	55500	23110
10	1	4	4	72060	26054
8	1	4	4	73680	26342
4	1	4	4	81774	27782
2	1	4	4	98160	30694
1	1	4	4	130920	36518
Refinement in width (z-direction)					
4	1	2	4	60672	23350
4	1	6	4	102888	32214
4	1	8	4	123996	36646
4	1	10	4	145104	41078
4	1	12	4	166212	45510
Refinement in height (y-direction)					
4	1	12	6	192876	52158
4	1	12	8	240648	63238
Final mesh selected					
4	1,2	12	4	151692	42630

Table E.1: Different mesh distributions used for the analysis of the convergence of the results.

In the next subsections the results are shown for all the cases studied.

Normal stress

Maximum normal stress is analysed in Figure E.8 .

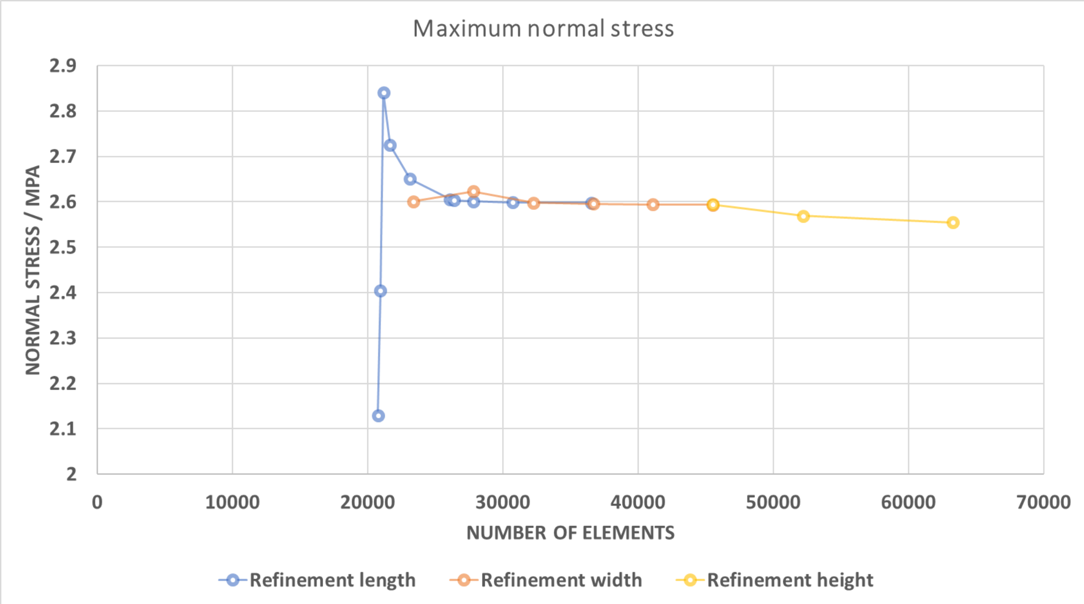


Figure E.8: Maximum normal stress at the interface for several mesh configurations.

As can be seen, refinement in all directions seem to have a similar impact on the convergence of the maximum normal stress at the interface. Note that even though the onset traction of the exponential law shown in Figure E.1 is of 2.5 MPa, all the meshes seem to converge to a value that is slightly over the onset traction defined. This is produced by the 3D effects and it is explained later in the report.

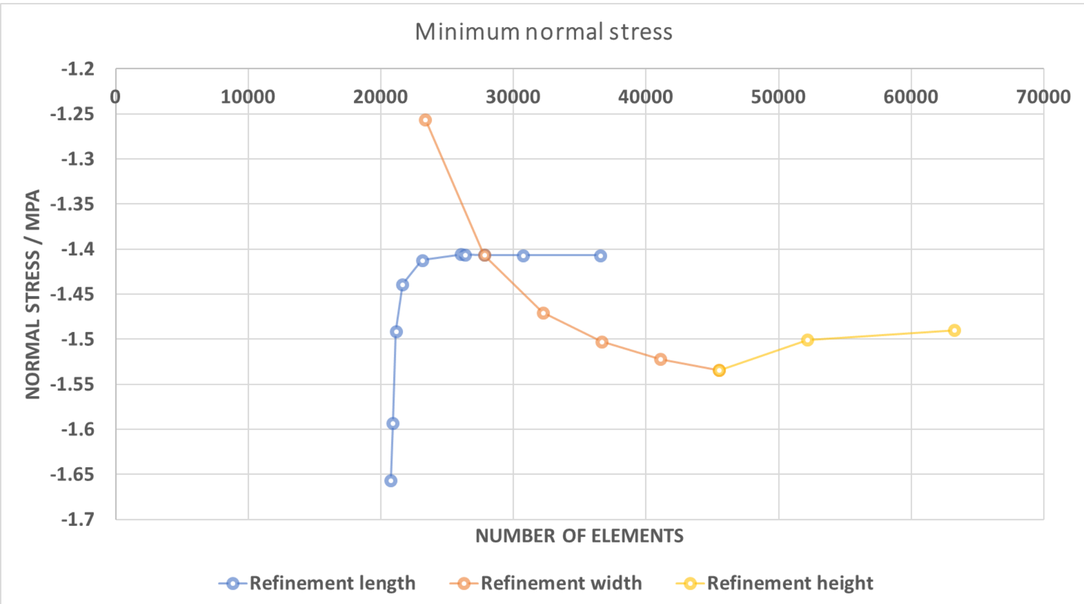


Figure E.9: Minimum normal stress at the interface for several mesh configurations.

Minimum normal stress results (Figure E.9) do converge after the refinement in the x-direction. However, when the width and height of the beam are continuously refined the initial "converged value" changes slowly until stabilizing in a different value. However, the change in value from the last point of the blue line with respect to the other lines is around 3%, so it can be considered converged.

Maximum displacement along the width

Maximum displacement value fluctuation along a path (located just before the crack tip at the interface), defined across the width of the DCB specimen close to the interface, is shown in Figure E.10.

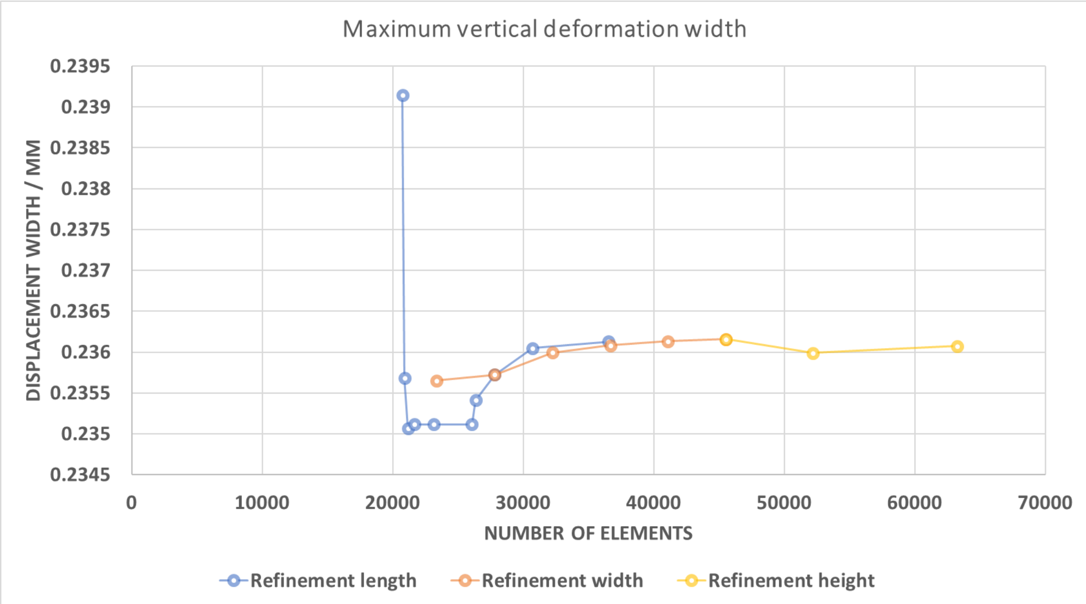


Figure E.10: Maximum displacement across the width of the DCB for several mesh configurations.

It is seen how convergence is achieved by a refinement across the length itself, refinement in other directions do not make a difference on the maximum value, but it is observed that it smoothens the displacement curve.

Regarding the total maximum deformation, convergence is also achieved only with the refinement in the length direction (Figure E.11).

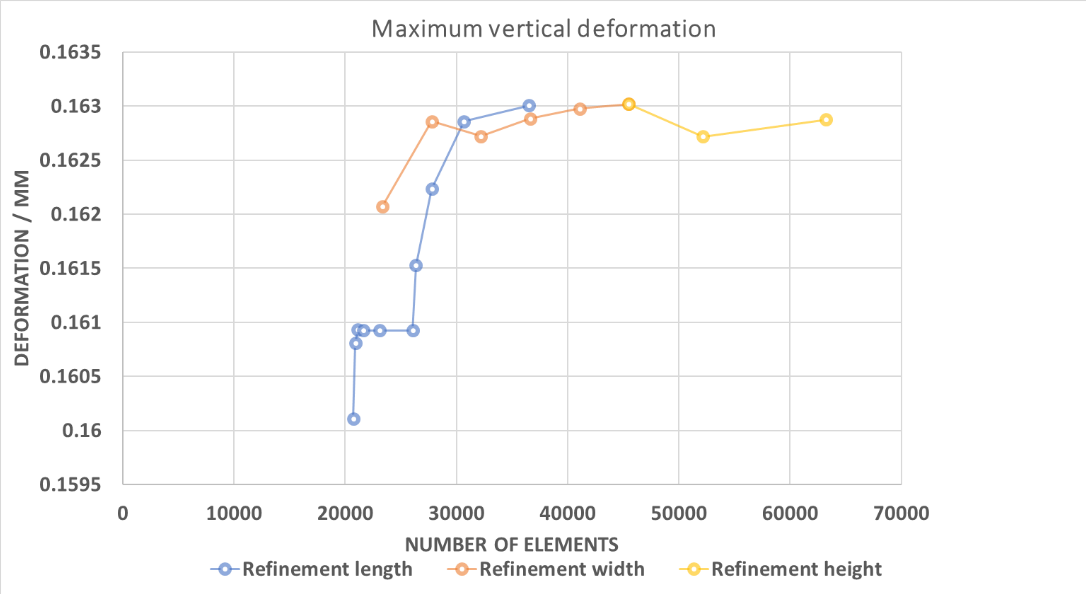


Figure E.11: Maximum vertical displacement of the DCB.

Force reactions

The force reaction in the force application areas are also subjected to the mesh convergence analysis (see Figures E.12).

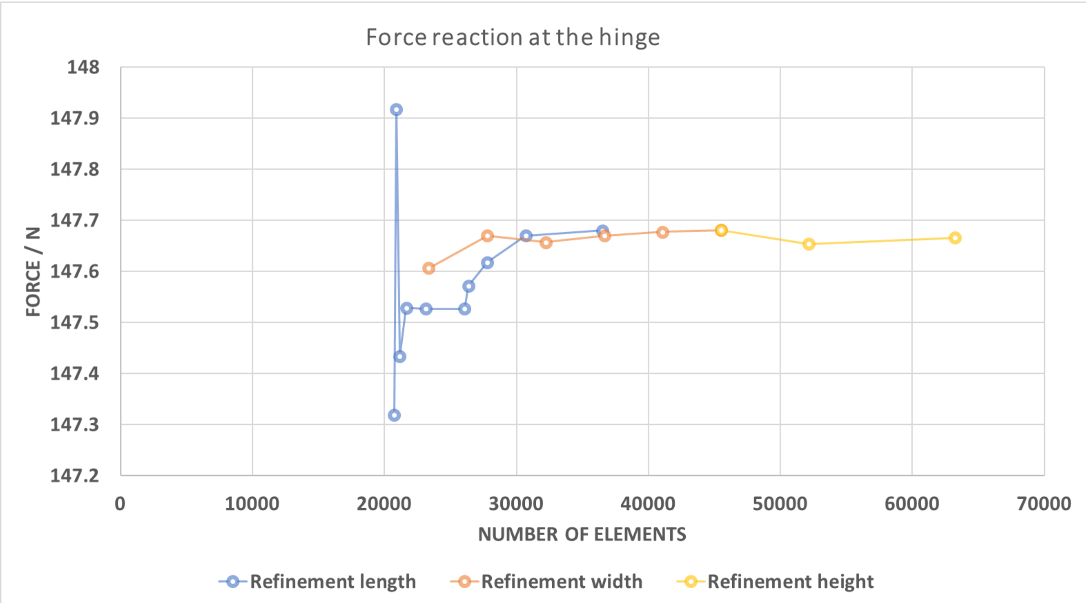


Figure E.12: Maximum force reaction at the hinges over the entire delamination process.

The convergence is seen with refinement along the length of the beam, once this value is reached, refinement in other directions have little influence on the value obtained.

Force convergence of the fixture is not shown in the project as the values are almost zero for all the simulations performed. Therefore, it is always converged.

E.2.1 Results and Final Mesh

There is a clear trend when examining the results obtained with the different mesh configurations used. The refinement on the edges of the height/width of the beam do not affect that much the results once convergence is obtained refining the length of the beam (the minimum normal stress is the only result that does not comply with this statement). This means, the focus should be on refining the elements along the length of the DCB specimen primarily. Then, refinement along the width is also desired due to the interest on 3D effects and due to the smoothing of the curves obtained. Finally, refinement on the height of the specimen is used to provide the elements a good aspect ratio, so that the behaviour of the elements can be as good as possible [Cook et al., 2002].

The mesh chosen for the final analysis has 42630 elements. The characteristics of the mesh chosen can be seen in the last row of Table E.1. The priority is to make the length and width as fine as possible, and to just adjust the height number of divisions to achieve the best mesh quality without using a high number of elements. An element size of 1.2 mm instead of 1 mm is used to reduce the distortion of the elements used (note that this change in element size does not represent any change in the results). The number of elements present in the cohesive zone is 5 elements, which is acceptable according to experience. The aspect ratio achieved is improved with these modifications made to the initial mesh distributions analysed in the previous paragraphs (Figure E.13). Note that tetrahedral elements are just used for the hinges, so they are not prioritized in the improvement of element quality.

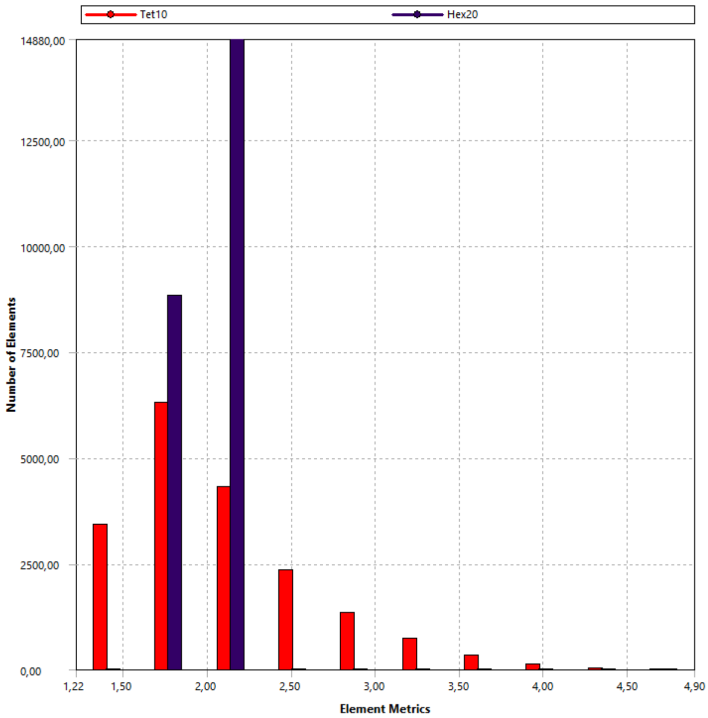


Figure E.13: Aspect ratio histogram of the final mesh.

F Uncertainty Evaluation of the Three-Point Bending Test

It is widely known that composite materials may present alterations in the values of their elastic properties, with respect to the nominal values, due to the fact that the fabrication is a manual process. Therefore, in order to be able to formulate the inverse parameter problem, the flexural rigidity and the shear rigidity must be obtained. The *flexural rigidity* EI, where E is the Young's modulus and I the moment of inertia, and the *shear rigidity* kAG, where k is the shear factor, A is the area of the cross section, and G the shear modulus.

In order to characterize these values and their uncertainty, it is decided to perform a three-point bending test of one of the arms of an already delaminated specimen. All the specimens provided are cut from the same plate, thus it is assumed that the properties do not vary significantly from one specimen to another. This test is chosen due to the ease of obtaining the desired values, and because the machines available at the workshop of the Department of Materials and Production allow for the preparation and performance of it. The test is to be carried out in the Zwick tensile test machine. Figure F.1 exemplifies the experiment where a vertical load, P, is applied at the midpoint of the specimen, that stands over two supports. The distance between the supports is defined as the span length L_s. It might be different from the length of the specimen, and it can be modified from one test to another. As the test sample is one arm of the DCB, its thickness h_s, is the half of the initial thickness stated in Table 3.1. This is due to the formulation of the beam-based model, where the height is the one from one arm of the DCB.

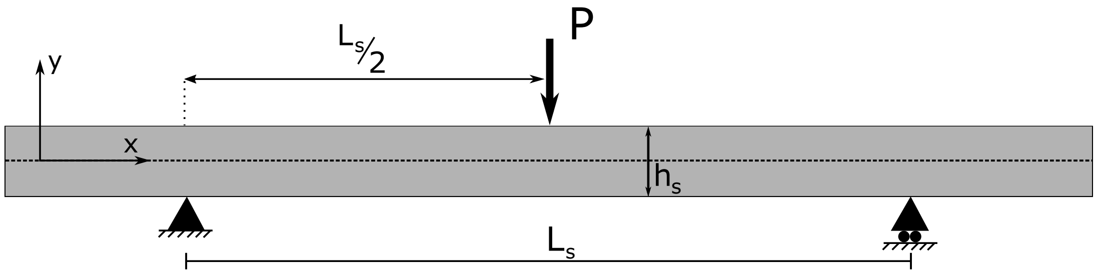


Figure F.1: Three-point bending diagram.

From the analysis of the setup, the flexural rigidity can be obtained from the expression of the maximum bending stress σ_M see Equation (F.1), where h_s is the thickness of the sample and L_s, the span length. The stress is expressed as the product of the Young's modulus E and the strain ε.

$$\sigma_M = E\varepsilon = \frac{-PL_s h_s}{8I} \quad \rightarrow \quad EI = \frac{-PL_s h_s}{8\varepsilon} \tag{F.1}$$

Therefore, if during the three-point bending test the force P and the strain ε are recorded, the flexural rigidity can be determined. Stated before, the test is performed with a tensile test machine, in which the force is being recorded during the process. In order to obtain the strain measurement, a strain gauge must be placed at the point where the bending stress equation is formulated i.e. at the bottom of the midpoint ($L_s/2$) of the specimen.

As the flexural rigidity is a function of other values (see Equation (C.1) in Appendix C), the law for accumulation of uncertainties must be utilized to characterize its uncertainty. The different parameters which contribute to the accumulated standard uncertainty are stated in the following.

- The force P, is measured via a load cell. In the case of the Zwick machine, the transducer is a HBM U9B, which has an accuracy class of 0.5. Following Mouritsen [2013], the uncertainty of the force measured $u(P)$, is a 0.5 % of the measured force.
- The strain ε, is measured with a strain gauge, and its associated uncertainty is mainly due to the gauge factor [Mouritsen, 2013]. This value is specified by the manufacturer, and it varies depending on the type of strain gauge. The absolute uncertainty of the strain $u(\varepsilon)$ is obtained from multiplying the measured strain by the relative uncertainty of the gauge factor $u(GF)$.
- The span length L_s, and the thickness h_s, are measured as with its former values from Table 3.1. Therefore, the type-B uncertainty is already known. As for the type-A uncertainty, it must be calculated again from a series of measurements. Thus, the uncertainties of the span length $u(L_s)$ and the thickness of the specimen $u(h_s)$ are known.

Now, by applying the law for accumulation of uncertainties, the expression for the uncertainty of the flexural rigidity can be obtained, and it is stated in the following equation.

$$u(EI) = \sqrt{\left(\frac{-L_s h_s}{8\varepsilon}u(P)\right)^2 + \left(\frac{-P h_s}{8\varepsilon}u(L_s)\right)^2 + \left(\frac{-P L_s}{8\varepsilon}u(h_s)\right)^2 + \left(\frac{P L_s h_s}{8\varepsilon^2}u(\varepsilon)\right)^2} \tag{F.2}$$

Following the approach presented in Muttashar et al. [2015], the value of the shear rigidity can be calculated if the vertical displacement δ of the midpoint of the specimen is known. The expression that relates the vertical displacement and the shear rigidity is obtained from the application of Timoshenko beam theory onto this particular problem. See Equation (F.3), wherein the flexural rigidity EI can be spotted. The other parameters in the expression are: the displacement δ, the force P, the span length of the beam L_s, and the shear rigidity kAG.

$$\delta = \frac{PL_s^3}{4EI} + \frac{PL_s}{48kAG} \tag{F.3}$$

Except from the vertical displacement, all the values that are present in the above expression already have its uncertainty characterized. The vertical displacement is measured via a dial gauge, placed in the same location as the strain gauge. The dial gauge has a resolution of 0.01 mm. There is one measurement per experiment, thus the uncertainty is evaluated as a type-B..

Now, with all the sources of uncertainty characterized, the accumulated standard uncertainty can be obtained from Equation (F.4).

$$u(kAG) = \Bigg\{ \Bigg[\Bigg(\frac{12EIL_s}{48EI\delta - L_s^3 P} + \frac{12EIL_s^4 P}{(48EI\delta - L_s^3 P)^2} \Bigg) u(P) \Bigg]^2 + \Bigg[\Bigg(\frac{12EIL_s}{48EI\delta - L_s^3 P} + \frac{12EIL_s^3 P^2}{(48EI\delta - L_s^3 P)^2} \Bigg) u(L_s) \Bigg]^2$$
$$+ \Bigg[-\frac{576(EI)^2 L_s P}{(48EI\delta - L_s^3 P)^2} u(\delta) \Bigg]^2 + \Bigg[\Bigg(\frac{12L_s P}{48ER\delta - L_s^3 P} + \frac{576EIL_s P\delta}{(48EI\delta - L_s^3 P)^2} \Bigg) u(EI) \Bigg]^2 \Bigg\}^{1/2} \tag{F.4}$$

G | Camera Calibration

Prior to the calibration procedure, some steps have to be taken to ensure the acquisition of trustworthy values. The camera is warmed up until a stable operating temperature is achieved. The time may vary from several minutes to hours depending on laboratory conditions and the camera model. For this particular project, the camera is turned on 30 minutes before taking any valid photos. If this step is not made there is a risk of introducing errors due to thermal expansion of the lens, camera or camera supports [Yu and Lubineau, 2019].

Camera calibration is done for determining the geometric and optical characteristics of the camera and/or the 3D position and rotation of the object with respect to the camera frame [Heikkila and Silven, 1997]. For this project the objective of the explicit camera calibration procedure is to determine the intrinsic parameters and distortion coefficients of the camera based on image observations of a checkerboard target. The intrinsic parameters of the camera are the focal length, scale factor and image center [Heikkila and Silven, 1997]. If images are not treated for distortion before the DIC correlation procedure, errors might be introduced in the displacement fields calculated [Jones and Iadicola, 2018].

The first step of the calibration procedure is to select the calibration target. In this case, a 7x9 checkerboard pattern with squares of 20mm size is printed and placed on to a wooden board (see Figure G.1).

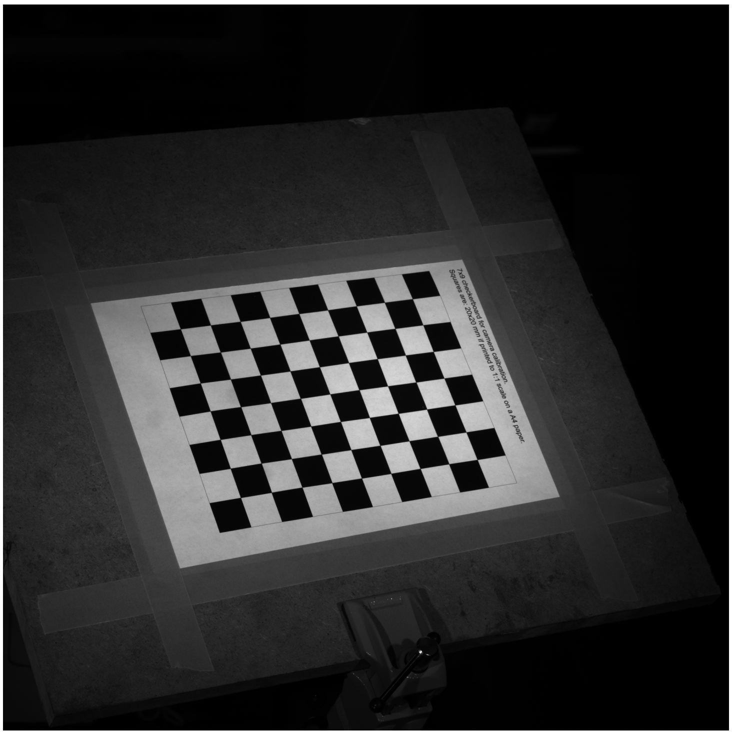


Figure G.1: Picture of the checkerboard pattern rotated with respect to the camera used for the calibration process.

A set of images with the checkerboard in different orientations and different positions with respect to the camera is taken (for a good calibration a minimum of 15 images is needed [Jones and Iadicola, 2018]). The calibration procedure is done using the MATLAB Camera Calibration app, included in the Computer Vision toolbox. This algorithm uses the pinhole model to make the calculations of the parameters (for more information about the formulation of the algorithm see Heikkila and Silven [1997] and Zhang [2000]). Images are introduced in this app and they are processed automatically when the square size is introduced as input. Out of 22 pictures taken, 18 are selected as valid by the algorithm and 4 of them are rejected (due to bad illumination or bad resolution). The options selected are standard camera, due to the type of lens that is used. For the calculations, 3 radial distortion coefficients are chosen for a better accuracy together with tangential distortion and skew.

The program locates the corners of each checkerboard pattern, and then makes the calculations to compute the image coordinates of the corners from the real world coordinates (see Figure G.2). The error between the detected corners (green circles in Figure G.2) and the projected corners (red crosses in Figure G.2) is denoted as *reprojection error*. The reprojection error is used as a qualitative measure of accuracy of the calibration method, and should be as close to 0 as possible.

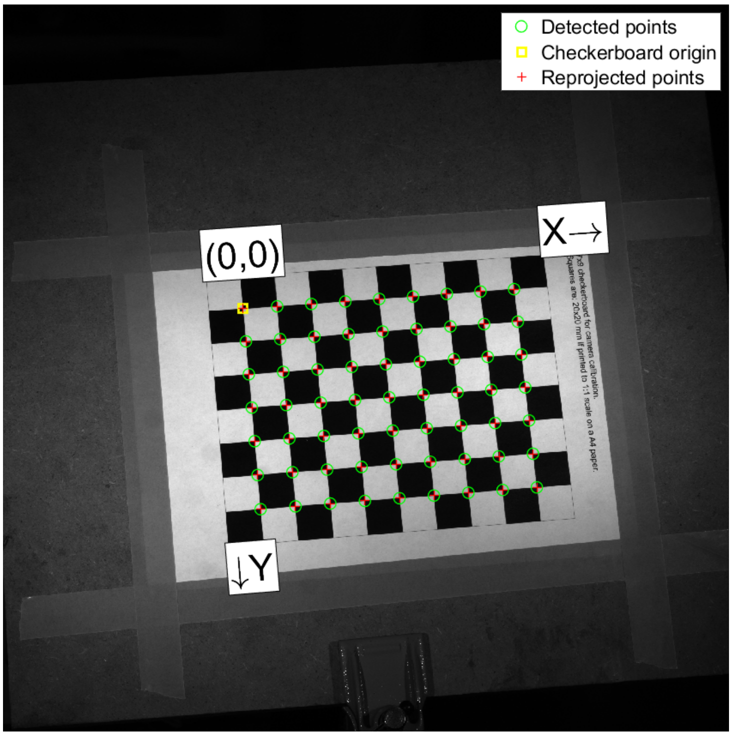


Figure G.2: Checkerboard pattern with the corners detected by the algorithm in green and reprojected points in red.

The mean reprojection error (Figure G.3) is calculated for each calibration image and the overall mean error is obtained.

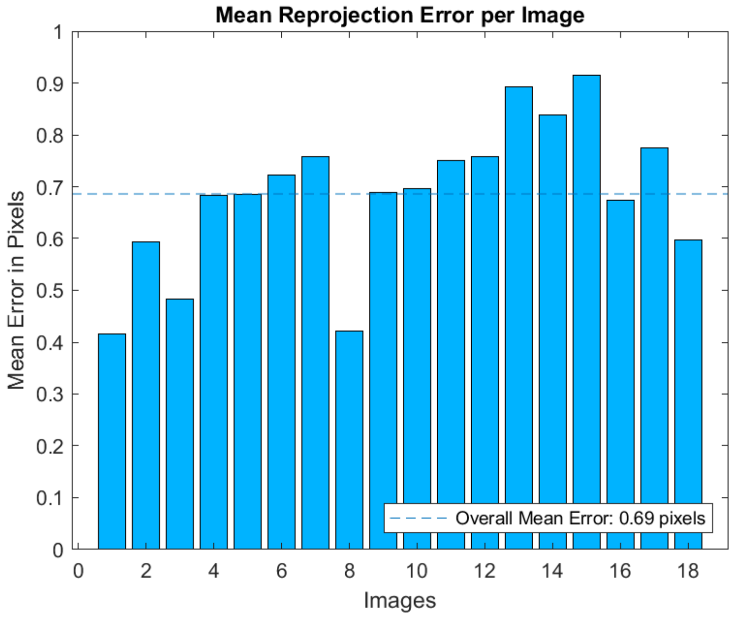


Figure G.3: Histogram showing the mean reprojection error calculated for each calibration image.

It can be seen how there is not really any image that exhibits an strange response. Therefore, it can be considered that the overall mean reprojection error is a good representation of the reality and the parameter calculations have been successful. The calculated positions of the pattern in the 3D space with respect to the camera can be seen in Figure G.4. This correspond to the calculation of the extrinsic parameters of the camera and it is used to check if any of the calculations is wrong.

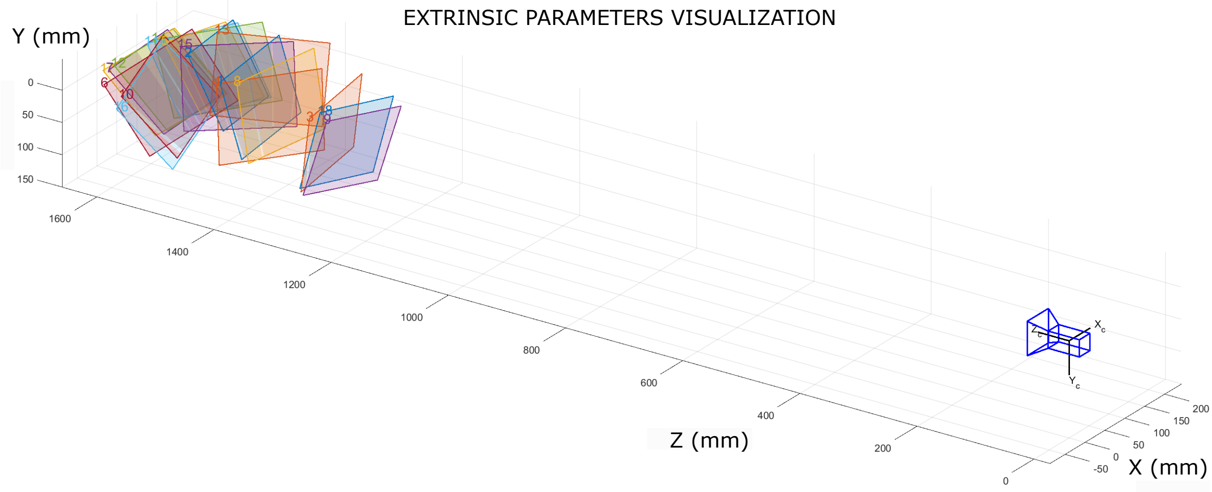


Figure G.4: Graph showing the positions and orientations calculated by the algorithm of the pattern with respect to the camera, for each calibration image.

Now the camera has been calibrated and all the parameters have been calculated. These are used to remove the distortion of the images before using them for the computation of the displacement field. After this process, the actual picture acquisition process takes place.

H Derivation of the constraint equations for the linear least squares.

The derivation of the constraints equations for the linear system of equations is shown here. For an explanation of the implementation of linear equality constraints to a linear least squares system of equations see Appendix B.

H.1 C^1-continuity

As it can be seen in Figure 6.11, the C^1-continuity is assured by the formulation at every point except where the peaks of the individual loads are located, i.e. at the point x_i, $\forall i \in [1, n]$. Thereafter, the objective is to restrain the slope of the resulting distributed load, the one obtained after summing all the individual load functions, to have the same value at both sides of these points.

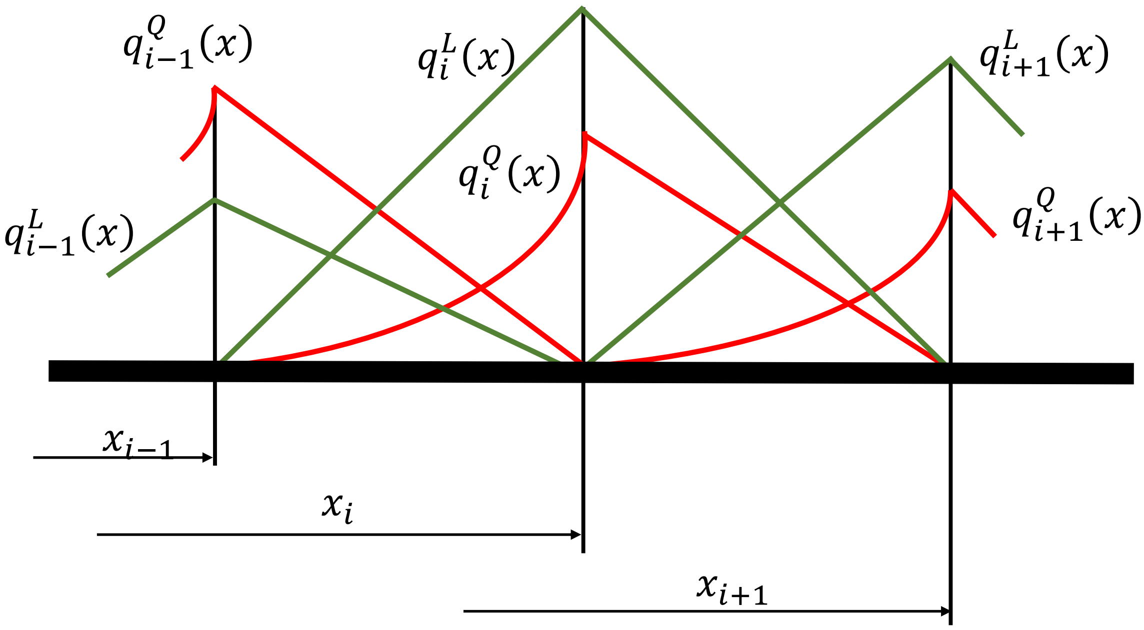


Figure H.1: Three triangular (green) and quadratic (red) loads with the position of its peaks x_i.

First, the slope at both sides for the first loading functions ($i = 1$), is calculated. See Equations (H.2) and (H.1).

$$\frac{dq}{dx}(x_i^-) = \frac{w_i^L}{x_i - x_{i-1}} + \frac{2w_i^Q}{x_i - x_{i-1}}, \quad i = 1 \tag{H.1}$$

$$\frac{dq}{dx}(x_i^+) = \frac{w_{i+1}^L}{x_{i+1}-x_i} - \frac{w_i^L}{x_{i+1}-x_i} - \frac{w_i^Q}{x_{i+1}-x_i}, \quad i=1 \tag{H.2}$$

where $\frac{dq}{dx}(x_i^-)$ is the derivative of q at a point on the left of x_i but infinitely close and $\frac{dq}{dx}(x_i^+)$ is the derivative of q at a point on the right of x_i but infinitely close.

Then, it is calculated for the load functions closest to the free end, $i = n$, in Equations (H.3) and (H.4).

$$\frac{dq}{dx}(x_i^-) = \frac{w_i^L}{x_i-x_{i-1}} - \frac{w_{i-1}^L}{x_i-x_{i-1}} - \frac{w_{i-1}^Q}{x_i-x_{i-1}} + \frac{2w_i^Q}{x_i-x_{i-1}}, \quad i=n \tag{H.3}$$

$$\frac{dq}{dx}(x_i^+) = -\frac{w_i^L}{x_{i+1}-x_i} - \frac{w_i^Q}{x_{i+1}-x_i}, \quad i=n \tag{H.4}$$

Finally, it is calculated for the rest of the load functions, see Equations (H.5) and (H.6).

$$\frac{dq}{dx}(x_i^-) = \frac{w_i^L}{x_i-x_{i-1}} - \frac{w_{i-1}^L}{x_i-x_{i-1}} - \frac{w_{i-1}^Q}{x_i-x_{i-1}} + \frac{2w_i^Q}{x_i-x_{i-1}}, \quad \forall i \in [2, n-1] \tag{H.5}$$

$$\frac{dq}{dx}(x_i^+) = \frac{w_{i+1}^L}{x_{i+1}-x_i} - \frac{w_i^L}{x_{i+1}-x_i} - \frac{w_i^Q}{x_{i+1}-x_i}, \quad \forall i \in [2, n-1] \tag{H.6}$$

Then, the same value of the slope at both sides is forced by equating the expression of the slope for each load, the constraint equation can be achieved.

$$(w_i^L + 2w_i^Q)K_i - (w_{i+1}^L) = 0, \quad K_i = \frac{x_{i+1}-x_i}{x_i-x_{i-1}}, \quad i=1 \tag{H.7}$$

$$(w_i^L + 2w_i^Q - w_{i-1}^L - w_{i-1}^Q)K_i - (w_{i+1}^L - w_i^L - w_i^Q) = 0, \quad K_i = \frac{x_{i+1}-x_i}{x_i-x_{i-1}}, \quad \forall i \in [2, n-1] \tag{H.8}$$

$$(w_i^L + 2w_i^Q - w_{i-1}^L - w_{i-1}^Q)K_i - (-w_i^L - w_i^Q) = 0, \quad K_i = \frac{x_{i+1}-x_i}{x_i-x_{i-1}}, \quad i=n \tag{H.9}$$

Arranging Equation (H.8) in matrix form, the constraint submatrices for the triangular distributed loads $Z_{C^1}^L \in \mathbb{R}^{n\times n}$, the quadratic distributed loads $Z_{C^1}^Q \in \mathbb{R}^{n\times n}$ and the force $Z_{C^1}^F \in \mathbb{R}^{n\times 1}$ can be formulated as in Equations (H.10), (H.11) and (H.12), respectively.

$$Z_{C^1}^L = \begin{bmatrix} (K_1+1) & -1 & & & & & \\ -K_2 & (K_2+1) & -1 & & & & \\ & -K_3 & \ddots & \ddots & & & \\ & & \ddots & \ddots & & & \\ & & & & -1 & & \\ & & & -K_{n-1} & (K_{n-1}+1) & -1 \\ & & & & -K_n & (K_n+1) \end{bmatrix} \tag{H.10}$$

$$Z_{C^1}^Q = \begin{bmatrix} (2K_1+1) & 0 & & & & & \\ -K_2 & (2K_2+1) & 0 & & & & \\ & -K_3 & \ddots & \ddots & & & \\ & & \ddots & \ddots & & & \\ & & & & 0 & & \\ & & & -K_{n-1} & (2K_{n-1}+1) & 0 \\ & & & & -K_n & (2K_n+1) \end{bmatrix} \tag{H.11}$$

$$Z^F_{C^1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{H.12}$$

The constraint equation for the C^1-continuity $Z_{C^1} \in \mathbb{R}^{n \times 2n+1}$ is achieved by concatenating the three previous matrices in Equations (H.10), (H.11) and (H.12).

$$Z_{C^1} = \begin{bmatrix} Z^L_{C^1} & Z^Q_{C^1} & Z^F_{C^1} \end{bmatrix} \tag{H.13}$$

$$Z_{C^1} \begin{bmatrix} w^L_1 & \dots & w^L_i & \dots & w^L_n & w^Q_1 & \dots & w^Q_i & \dots & w^Q_n & F \end{bmatrix}^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{H.14}$$

There are n C^1-continuity constraint equations. Therefore, after introducing the C^1 continuity constraint to the combination of triangular and quadratic load functions, the effective number of traction d.o.f. is the same as using only triangular load functions without constraints.

H.2 Force Equilibrium Constraint

The force equilibrium is imposed by calculating the sum of forces in the vertical direction and equating it to zero, see Equation (H.15).

$$F + \int_0^L q(x)dx = F + \sum_{i=1}^{n} \int_0^L \left(w^L_i q^L_i(x) + w^Q_i q^Q_i(x) \right) dx = 0 \tag{H.15}$$

The contribution of each load is calculated as

$$\int_0^L q^L_i(x)dx = w^L_i \left(\frac{x_{i+1} - x_{i-1}}{2} \right) \tag{H.16}$$

$$\int_0^L q^Q_i(x)dx = w^Q_i \left(\frac{x_{i+1} - x_{i-1}}{3} + \frac{x_{i+1} - x_i}{2} \right) \tag{H.17}$$

The submatrices for the triangular load functions $Z^L_{Feq} \in \mathbb{R}^{1 \times n}$, the quadratic load functions $Z^Q_{Feq} \in \mathbb{R}^{1 \times n}$ and the force $Z^F_{Feq} \in \mathbb{R}^{1 \times 1}$ are displayed in Equations (H.18), (H.19) and (H.20), respectively.

$$Z^L_{Feq} = \begin{bmatrix} \frac{x_2 - x_0}{2} & \dots & \frac{x_{i+1} - x_{i-1}}{2} & \dots & \frac{x_{n+1} - x_{n-1}}{2} \end{bmatrix} \tag{H.18}$$

$$Z^Q_{Feq} = \begin{bmatrix} \left(\frac{x_2 - x_0}{3} + \frac{x_2 - x_1}{2} \right) & \dots & \left(\frac{x_{i+1} - x_{i-1}}{3} + \frac{x_{i+1} - x_i}{2} \right) & \dots & \left(\frac{x_{n+1} - x_{n-1}}{3} + \frac{x_{n+1} - x_n}{2} \right) \end{bmatrix} \tag{H.19}$$

$$Z^F_{Feq} = \begin{bmatrix} 1 \end{bmatrix} \tag{H.20}$$

The constraint matrix for enforcing force equilibrium is displayed in Equation (H.22).

$$Z_{Feq} = \begin{bmatrix} Z^L_{Feq} & Z^Q_{Feq} & Z^F_{Feq} \end{bmatrix} \tag{H.21}$$

$$Z_{Feq} \begin{bmatrix} w^L_1 & \dots & w^L_i & \dots & w^L_n & w^Q_1 & \dots & w^Q_i & \dots & w^Q_n & F \end{bmatrix}^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{H.22}$$

H.3 Moment Equilibrium Constraint

In order to ensure moment equilibrium, the contribution of the moments generated around the clamp by each load function, $q_i^L(x)$ and $q_i^Q(x)$, and the force F must add to zero, see Equation (H.23).

$$FL + \int_0^L q(x)xdx = FL + \sum_{i=1}^{n} \int_0^L \left(w_i^L q_i^L(x)x + w_i^Q q_i^Q(x)x \right) dx = 0 \tag{H.23}$$

The moment generated by each load is calculated as

$$w_i^L \int_0^L q_i^L(x)xdx = w_i^L \left(\frac{(x_i - x_{i-1})(2x_i + x_{i-1}) + (x_{i+1} - x_i)(x_{i+1} + 2x_i)}{6} \right) \tag{H.24}$$

$$w_i^Q \int_0^L q_i^Q(x)xdx = w_i^Q \left(\frac{(x_i - x_{i-1})(3x_i + x_{i-1}) + 2(x_{i+1} - x_i)(x_{i+1} + 2x_i)}{12} \right) \tag{H.25}$$

The submatrices for the triangular load functions $Z_{Meq}^L \in \mathbb{R}^{1\times n}$, the quadratic load functions $Z_{Meq}^Q \in \mathbb{R}^{1\times n}$ and the force $Z_{Meq}^F \in \mathbb{R}^{1\times 1}$ are displayed in Equations (H.18), (H.19) and (H.20).

$$\begin{aligned} Z_{Meq}^L = \Big[&\left(\tfrac{(x_1-x_0)(2x_1+x_0)+(x_2-x_1)(x_2+2x_1)}{6} \right) \quad \dots \quad \left(\tfrac{(x_i-x_{i-1})(2x_i+x_{i-1})+(x_{i+1}-x_i)(x_{i+1}+2x_i)}{6} \right) \\ &\dots \quad \left(\tfrac{(x_n-x_{n-1})(2x_n+x_{n-1})+(x_{n+1}-x_n)(x_{n+1}+2x_n)}{6} \right) \Big] \end{aligned} \tag{H.26}$$

$$\begin{aligned} Z_{Meq}^Q = \Big[&\left(\tfrac{(x_1-x_0)(3x_1+x_0)+2(x_2-x_1)(x_{i+1}+2x_i)}{12} \right) \quad \dots \quad \left(\tfrac{(x_i-x_{i-1})(3x_i+x_{i-1})+2(x_{i+1}-x_i)(x_{i+1}+2x_i)}{12} \right) \\ &\dots \quad \left(\tfrac{(x_i-x_{i-1})(3x_i+x_{i-1})+2(x_{i+1}-x_i)(x_{i+1}+2x_i)}{12} \right) \Big] \end{aligned} \tag{H.27}$$

$$Z_{Meq}^F = \begin{bmatrix} L \end{bmatrix} \tag{H.28}$$

The constraint matrix of the moment equilibrium constraint $Z_{Meq} \in \mathbb{R}^{1\times 2n+1}$ is obtained by concatenating the submatrices in Equations (H.26), (H.27) and (H.28).

$$Z_{Meq} = \begin{bmatrix} Z_{Meq}^L & Z_{Meq}^Q & Z_{Meq}^F \end{bmatrix} \tag{H.29}$$

$$Z_{Meq} \begin{bmatrix} w_1^L & \dots & w_i^L & \dots & w_n^L & w_1^Q & \dots & w_i^Q & \dots & w_n^Q & F \end{bmatrix}^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{H.30}$$

H.4 Zero Slope at the Point Closest to the Clamp

In Figure H.1, it can be seen that the only loading function which provides a slope different than zero at the closest point to the clamp x_1, is the linear load $q_i^L(x)$. Therefore, the constraint which ensures zero slope at x_1 is

$$w_i^L = 0 \tag{H.31}$$

which is written in Equation (H.33) in matrix form.

$$Z_{slp1} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \tag{H.32}$$

$$Z_{slp1}\begin{bmatrix} w_1^L & \dots & w_i^L & \dots & w_n^L & w_1^Q & \dots & w_i^Q & \dots & w_n^Q & F \end{bmatrix}^T = 0 \tag{H.33}$$

This formulation might seen useless, as it can be accomplished by simply removing the first column of the compliance matrix of the triangular load functions $G^L(x)$. However, the formulation presented here allows for a more flexible *MATLAB* routine.

H.5 Zero Slope at the Point Closest to the Free End

The slope of at the point closest to the free end x_n, is the sum of the slope w_n^L and w_n^Q. Therefore, the constraint equation is

$$w_n^L + w_n^Q = 0 \tag{H.34}$$

which in matrix form becomes

$$Z_{slpn} = \begin{bmatrix} 0 & \dots & 0 & \dots & 1 & 0 & \dots & 0 & \dots & 1 & 0 \end{bmatrix} \tag{H.35}$$

$$Z_{slpn}\begin{bmatrix} w_1^L & \dots & w_i^L & \dots & w_n^L & w_1^Q & \dots & w_i^Q & \dots & w_n^Q & F \end{bmatrix} = 0 \tag{H.36}$$

H.6 Final Constraints Matrix *Z*

Finally, the matrix constraint including all the explained constraints is built concatenating the already defined matrices.

$$Z = \begin{bmatrix} Z_{C^1} \\ Z_{Feq} \\ Z_{Meq} \\ Z_{slp1} \\ Z_{slpn} \end{bmatrix} \tag{H.37}$$

I | MATLAB files

This appendix contains the MATLAB files relevant in this project.

I.1 Beam Displacement Analytic Derivation Function

Function which derives the compliance function of a given load function $q(x, a, b)$, with symbolic variables. Where a and b are the limits of the distributed load and x is the coordinate which expressed the position on the beam.

```
function [ComplianceFun Disp1 Disp2 Disp3] = ComplianceFunDerivatorT(q)
%% Assumptions
syms E I L a b x k G A
assume([E I L b],{'Real','Positive'})
assumeAlso(0<=a<b<=L)
assumeAlso(0<=x<=L)
% Load function: this function is only defined between a and b (a<=x<=b),
% out of this interval is equal to zero
%% Calculations for Euler-Bernoulli
Vr = simplify(int(q,x,a,b));
Mr = simplify(int(q*x,x,a,b));
V(x) = -simplify(Vr - int(q,x,a,x));
M(x) = simplify(Mr - a*Vr + int(V,x,a,x));
Rot(x) = simplify(Vr*a^2/E/I/2 + (Mr-a*Vr)*a/E/I + int(M,x,a,x)/E/I);
d1(x) = -simplify(Vr*x^2*(3*a-x)/6/E/I + (Mr-Vr*a)*x^2/2/E/I);
d2(x) = -simplify(Vr*a^3/3/E/I + (Mr-a*Vr)*a^2/2/E/I + int(Rot,x,a,x),'Steps',100);
Rot2(x) = diff(d2,x);
d3(x) = -simplify(-Rot2(b)*(x-b)-d2(b));

%% Recalculation for Timoshenko
eb2timo = @(eb) int(diff(eb,x) - E*I/k/G/A*diff(eb,x,3));
Disp1 = eb2timo(d1);
Disp2 = eb2timo(d2);
Disp2 = Disp2 - Disp2(a) + Disp1(a);
Disp3 = eb2timo(d3);
Disp3 = Disp3 - Disp3(b) + Disp2(b);
ComplianceFun(x,a,b) = simplify(piecewise(x<a,Disp1,a<=x<=b,Disp2,x>b,Disp3),'Steps',100);
end
```

I.2 Inverse Parameter Identification Program

I.2.1 Main Script for Inverse Parameter Identification

Script which runs and postprocesses the InTraFiCa results.

```
%%%%% Program of inverse parameter identification of cohesive zone
%%%%% laws from experimental data of DIC. This program calculates the
%%%%% traction field using the displacement data from different locations
%%%%% of a 3D FE-model. The out-of-plane motion on the lateral face can be
%%%%% simulated.
```

```
%%%%%
%%%%% Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%%%%% Master in Design of Mechanical Systems, Aalborg University, Denmark.
clc;close all
restoredefaultpath

%% User inputs
% BB-model, specified as the folder containing the routines
addpath Routines_quad

% Calculation
SubsetSize = 15;%Existant: 9 11 13 15
steps2calculate = 1:51;% There are 51, input must be a vector of the pictures
% handle function with the options of InTraFiCa
IPIfunction = @(PD,disp) InTraFiCa(PD,disp,'NumberLoads',4,'TracLoc',[0.416 0.61]);

%Plotting
onlyplotting = false;% true if no calculation should be done, only plotting
toplot = [0];% steps to plot, toplot=0 means plot all
ExtraTitle = [''];

%% Problem data
ProblemData.h = 0.00425;
ProblemData.w = 0.0278;
ProblemData.kAG = 5/6*2.2e9*ProblemData.h*ProblemData.w;
ProblemData.EI = 4.595e10*ProblemData.h^3*ProblemData.w/12;
ProblemData.L = 0.64;

%% Importing DIC data
addpath DICdata
load(['DisplacementData_Size' num2str(SubsetSize) '_Test_long2'])
if length(steps2calculate)>length(DisplacementData)
    error(['The variable steps2calculate has more elements than the DIC results.'])
end

DisplacementData = DisplacementData(steps2calculate);
DisplacementDataPlot = DisplacementData;

%% Inverse parameter identification
if ~onlyplotting
    clearvars -except DisplacementData DisplacementDataPlot ExtraTitle ...
        toplot onlyplotting SubsetSize steps2calculate IPIfunction ProblemData
    num = length(steps2calculate);
    disp([num2str(num) ' calculation(s)...'])

    if num > 10
        parfor ii = 1:num
            BB_IPI(ii) = IPIfunction(ProblemData,DisplacementData{ii});
        end
    else
        for ii = 1:num
            BB_IPI(ii) = IPIfunction(ProblemData,DisplacementData{ii});
            disp([num2str(ii/num*100,3) '% completed'])
        end
    end
end

%% Postprocessing
if max(toplot)>length(steps2calculate)
    error(['The variable toplot has steps which has not been calculated.'...
```

```
        ' Check that the elements in toplot are in steps2calculate.'])
end
if toplot == 0
    toplot = 1:num;
end

for ii = 1:length(toplot)
    legendcell(ii) = {['Picture ' num2str(steps2calculate(toplot(ii)))]};
    legendcelldisp(ii) = {['Picture ' num2str(steps2calculate(toplot(ii)))]};
end

%Tractions plot
figure
yyaxis left
if ~isfield(BB_IPI,'CalculatedTractionsOrg')
    for ii = toplot
        plotlist(ii) =
            plot(BB_IPI(ii).TractionMesh,-BB_IPI(ii).CalculatedTractions/ProblemData.w,'LineWidth',1.5);
        hold on
    end
    for ii = toplot
        plot(BB_IPI(ii).TractionMesh,-BB_IPI(ii).CalculatedTractions/ProblemData.w,'o')
        hold on
    end
else
    for ii = toplot
        plotlist(ii) =
            plot(BB_IPI(ii).TractionMesh,-BB_IPI(ii).CalculatedTractions/ProblemData.w,'LineWidth',1.5);
        hold on
    end
end
hold off
xlabel('Position (m)')
title(['Interfacial tractions' ExtraTitle])
grid on
ylabel('Tractions (Pa)')

tmp = gca;
tmp = tmp.YLim;

yyaxis right
for ii = toplot
    plotlistdisp(ii) = plot(DisplacementData{ii}(:,1),DisplacementData{ii}(:,2));
    hold on
end
hold off
tmp2 = gca;
axis([tmp2.XLim -tmp2.YLim(2)*(diff(tmp)/tmp(2)-1) tmp2.YLim(2)])

h = legend([plotlist(toplot) plotlistdisp(toplot)],[legendcell legendcelldisp]);
h.NumColumns=2;
h.Location='northwest';

%Cohesive Law plot
figure
for ii = 1:num
    [CL(ii).CohesiveLaw CL(ii).RealPoints] = CLextracter(BB_IPI(ii),DisplacementDataPlot{ii});
end
cont = 0;
for ii = toplot
```

```
    if ~isempty(CL(ii).CohesiveLaw)
        plotlist2(ii) = plot(CL(ii).CohesiveLaw(:,1), CL(ii).CohesiveLaw(:,2),'-','LineWidth',1.5);
        hold on
    end
end
hold off
grid on
ylabel('Traction (Pa)');
xlabel('Crack opening (m)');
title(['Cohesive Laws' ExtraTitle]);
h = legend(plotlist2(toplot),legendcell);

% Other parameters
Step = steps2calculate';
if isfield(BB_IPI,'nonlin')
    NLstruct = [BB_IPI(:).nonlin];
    Exit_Flag = [NLstruct(:).exitflag]';
end
for ii = 1:num
    Force(ii,1) = BB_IPI(ii).ProblemData.F;
    Length(ii,1) = BB_IPI(ii).ProblemData.L;
    if isempty(CL(ii).CohesiveLaw)
        Onset_Traction(ii,1) = NaN;
        Final_Separation(ii,1) = NaN;
        Energy_RR(ii,1) = NaN;
    else
        [Onset_Traction(ii,1) postmp] = max(CL(ii).CohesiveLaw(:,2));
        Final_Separation(ii,1) = CL(ii).CohesiveLaw(end,1);
        Energy_RR(ii,1) = trapz(CL(ii).CohesiveLaw(:,1),CL(ii).CohesiveLaw(:,2));
    end
end

if isfield(BB_IPI,'nonlin')
    NLstruct = [BB_IPI(:).nonlin];
    Exit_Flag = [NLstruct(:).exitflag]';
    table(Step,Force,Length,Onset_Traction,Final_Separation,Energy_RR,Exit_Flag)
else
    table(Step,Force,Length,Onset_Traction,Final_Separation,Energy_RR)
end

figure
yyaxis left
plot(steps2calculate(toplot),Energy_RR(toplot),'o-')
ylabel('Energy Realease Rate (J/m^2)')
yyaxis right
plot(steps2calculate(toplot),Onset_Traction(toplot),'o-')
xlabel('Picture number')
ylabel('Onset Traction (Pa)')
title(['Variation along the delamination' ExtraTitle])

figure
cont = 0;
clearvars peakpos cracktip initialcomp
for ii = toplot %()
    cont = cont + 1;
    peakpos(cont) = BB_IPI(ii).TractionMesh(max(BB_IPI(ii).CalculatedTractions)...
        ==BB_IPI(ii).CalculatedTractions);
    if ~isfield(BB_IPI,'CalculatedTractionsOrg')
        postmp = max(find(-BB_IPI(ii).CalculatedTractions/ProblemData.w ...
            == CL(ii).RealPoints(end,2)));
```

```
        if postmp == length(BB_IPI(ii).CalculatedTractions)
           postmp = postmp-1;
        end
        cracktip(cont) = -diff(BB_IPI(ii).CalculatedTractions(postmp:postmp+1))...
            /BB_IPI(ii).CalculatedTractions(postmp)*...
            diff(BB_IPI(ii).TractionMesh(postmp-1:postmp))+BB_IPI(ii).TractionMesh(postmp-1);
        initialcomp(cont) = BB_IPI(ii).TractionMesh(1);
    else
        cracktip(cont) = BB_IPI(ii).TractionMeshOrg(end-1);
        initialcomp(cont) = BB_IPI(ii).TractionMeshOrg(2);
    end
end
plot(steps2calculate(toplot),[peakpos;cracktip;initialcomp],'-o')
xlabel('Picture number')
ylabel('Position (m)')
title(['Limits of the FPZ' ExtraTitle])
legend('Traction peak','Crack tip','Intial compression')
grid on

figure
plot(steps2calculate(toplot),Force(toplot),'o-')
ylabel('Delamination force (N)')
xlabel('Picture number')
grid on
```

I.2.2 ComplianceFunLin

Function which creates the compliance matrix of the triangular distributed loads used in the inverse parameter identification routine for a given set of load positions and beam data.

```
function BB_IPI = CompFunLin(BB_IPI) %()
%   COMPFUNLIN calculates the compliance matrix corresponding to the
%   triangular loading functions.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Input check
x = BB_IPI.DispMesh;
if size(BB_IPI.TractionMesh,1)~=1
    BB_IPI.TractionMesh = BB_IPI.TractionMesh';
end
if size(x,1)==1
    x = x';
end
a = BB_IPI.TractionMesh(1:end-2);%Limits of the area where the load is aplied (a<b<c)
b = BB_IPI.TractionMesh(2:end-1);%The peak is on b and the value in a and c is zero
c = BB_IPI.TractionMesh(3:end);
E = BB_IPI.ProblemData.EI;%This modification has been done because we are
% going to measure the flexural rigidity instead, the same with kAG
I = 1;%BB_IPI.ProblemData.I;
L = BB_IPI.ProblemData.L;
k = BB_IPI.ProblemData.kAG;
G = 1;%BB_IPI.ProblemData.G;
A = 1;%BB_IPI.ProblemData.h*BB_IPI.ProblemData.w;

%% Calculation
t2 = a.^2;
```

```
t3 = a.^3;
t5 = b.^2;
t6 = b.^3;
t8 = c.^2;
t9 = c.^3;
t10 = x.^2;
t11 = x.^3;
t13 = x.^5;
t14 = 1.0./A;
t15 = 1.0./E;
t16 = 1.0./G;
t17 = 1.0./I;
t18 = -b;
t19 = -c;
t20 = 1.0./k;
t21 = E.*I.*b.*4.0e+1;
t4 = t2.^2;
t7 = t5.^2;
t12 = t10.^2;
t22 = a+t18;
t23 = b+t19;
t24 = A.*G.*t6.*4.0;
t25 = A.*G.*t5.*x.*1.5e+1;
t26 = 1.0./t23;
t27 = -t25;
BB_IPI.CompMat = -((x <
    a).*((t14.*t15.*t16.*t17.*t20.*x.*(a+t19).*(E.*I.*6.0-A.*G.*k.*t10+A.*G.*a.*k.*x+A.*G.*b.*k.*x+A.*G.*c.*k.*x))./1.2e+1)...
+ ((a <= x) & (x <=
    b)).*((t14.*t15.*t16.*t17.*t20.*(E.*I.*t3.*2.0e+1-E.*I.*t11.*2.0e+1-A.*G.*a.^5.*k+E.*I.*a.*t10.*6.0e+1+A.*G.*k.*t13-A.*G.*a
+ ((b < x) & (x <=
    c)).*((t14.*t16.*t20.*(t2-t5.*2.0+a.*b))./6.0-(t15.*t17.*t26.*(t13-c.*t7.*5.0-c.*t12.*5.0+t8.*t11.*1.0e+1-t9.*t10.*1.0e+1-t
+ (c <
    x).*(t14.*t15.*t16.*t17.*t22.*(t24+t27+A.*G.*t3+A.*G.*a.*t5.*3.0+A.*G.*b.*t2.*2.0-A.*G.*t2.*x.*5.0-A.*G.*a.*b.*x.*1.0e+1).*

end
```

I.2.3 ComplianceFunQuadF

Function which creates the compliance matrix of the quadratic distributed loads used in the inverse parameter identification routine for a given set of load positions and beam data.

```
function BB_IPI = CompFunQuadF(BB_IPI)
%   COMPFUNQUADF calculates the compliance matrix corresponding to the
%   quadratic loading functions and the external force.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.
x = BB_IPI.DispMesh;
if size(BB_IPI.TractionMesh,1)~=1
    BB_IPI.TractionMesh = BB_IPI.TractionMesh';
end
if size(x,1)==1
    x = x';
end
a = BB_IPI.TractionMesh(1:end-2);%Limits of the area where the load is aplied (a<b<c)
b = BB_IPI.TractionMesh(2:end-1);%The peak is on b and the value in a and c is zero
c = BB_IPI.TractionMesh(3:end);
E = BB_IPI.ProblemData.EI;%This modification has been done because we are
% going to measure the flexural rigidity instead, the same with kAG
I = 1;%BB_IPI.ProblemData.I;
L = BB_IPI.ProblemData.L;
k = BB_IPI.ProblemData.kAG;
G = 1;%BB_IPI.ProblemData.G;
A = 1;%BB_IPI.ProblemData.h*BB_IPI.ProblemData.w;

t2 = a.^2;
t3 = a.^3;
t5 = b.*2.0;
t6 = b.^2;
t7 = b.^3;
t9 = c.^2;
t10 = c.^3;
t11 = x.^2;
t12 = x.^3;
t14 = x.^5;
t16 = 1.0./A;
t17 = 1.0./E;
t18 = 1.0./G;
t19 = 1.0./I;
t20 = -b;
t21 = -c;
t22 = 1.0./k;
t33 = a.*b.*x.*1.8e+1;
t4 = t2.^2;
t8 = t6.^2;
t13 = t11.^2;
t15 = a.*t5;
t23 = t6.*3.0;
t24 = c+t5;
t26 = t7.*1.0e+1;
t27 = a+t20;
t28 = b.*t2.*3.0;
t29 = a.*t6.*6.0;
t30 = b+t21;
t32 = t2.*x.*6.0;
```

```
t35 = t6.*x.*3.6e+1;
t36 = -t33;
t25 = -t23;
t31 = c.*t23;
t34 = -t32;
t37 = -t35;
t38 = 1.0./t27.^2;
t39 = 1.0./t30;
t41 = (t12.*t17.*t19.*t30)./1.2e+1;
t43 = (t16.*t18.*t22.*t30.*x)./2.0;
t44 = (t11.*t17.*t19.*t24.*t30)./1.2e+1;
t40 = t2+t15+t25;
t42 = -t41;
t46 = t3+t26+t28+t29+t34+t36+t37;
t45 = (t16.*t18.*t22.*t40)./1.2e+1;
t47 = (t17.*t19.*t27.*t46)./3.6e+2;
t48 = -t47;
BB_IPI.CompMat = -((x < a).*...
    (t42+t43+t44-(t12.*t17.*t19.*t27)./1.8e+1+(t11.*t17.*t19.*t27.*(a.*3.0+b.*9.0))./7.2e+1+(t16.*t18.*t22.*t27.*x)./3.0)...
+ ((a <= x) & (x < b)).*...
    (t42+t43+t44-(t17.*t19.*t38.*(a.*t14.*-6.0+t2.*t13.*1.5e+1-t7.*t12.*2.0e+1+t8.*t11.*4.5e+1-a.^5.*x.*6.0+t2.^3+t11.^3+a.*t6.*
+ (b == x).*...
    (t16.*t17.*t19.*t22.*(t11.*((A.*k.*t2)./4.0e+1-(A.*k.*t9)./1.2e+1)-(A.*k.*t4)./3.6e+2+(A.*k.*t13)./9.0e+1+A.*a.*k.*t12.*(7.0
+ ((b < x) & (x <= c)).*...
    (t45+t48-(t17.*t19.*t39.*(t14-c.*t8.*5.0-c.*t13.*5.0+t9.*t12.*1.0e+1-t10.*t11.*1.0e+1-t8.*x.*1.5e+1+b.^5.*4.0+c.*t7.*x.*2.0e
+ (c < x).*...
    (t45+t48-(t17.*t19.*t30.*(t7.*4.0+t10+t31+t5.*t9-t6.*x.*1.5e+1-t9.*x.*5.0-b.*c.*x.*1.0e+1))./1.2e+2-(t16.*t18.*t22.*(t6.*-2.
BB_IPI.CompMat = [BB_IPI.CompMat x.^2.*(3*L-x)/6/E/I+x/k/A/G];
end
```

I.2.4 InvParamId

This script calls the function *ComplianceFunLin* and *ComplianceFunQuad* and performs the inverse parameter identification of the tractions for a given set of displacements.

```
function [BB_IPI] = InvParamId(BB_IPI)
%   INVPARAMID finds a solution for the linear least-squares system.
%   Several algorithms are available.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Compliance matrix calculation
BB_IPI = BB_IPI.ComplianceFun(BB_IPI);

%% Input check
if size(BB_IPI.CompMat,1)~=length(BB_IPI.GivenDisplacements)
    BB_IPI.CompMat = BB_IPI.CompMat.';
end
if size(BB_IPI.GivenDisplacements,1)~=length(BB_IPI.GivenDisplacements)
    BB_IPI.GivenDisplacements = BB_IPI.GivenDisplacements.';
end

%% Algorithm selection
switch BB_IPI.LinAlgorithm
    case 'pseudoinverse'
        BB_IPI.CalculatedTractions =
             inv(BB_IPI.CompMat.'*BB_IPI.CompMat)*BB_IPI.CompMat.'*BB_IPI.GivenDisplacements;%Least-squares
             formula
    case 'MP'
        BB_IPI.CalculatedTractions = pinv(BB_IPI.CompMat)*BB_IPI.GivenDisplacements;%Least-squares formula
    case 'QR'
        BB_IPI.CalculatedTractions = BB_IPI.CompMat\BB_IPI.GivenDisplacements;%Least-squares formula
    case 'lsqminnorm'
        BB_IPI.CalculatedTractions = lsqminnorm(BB_IPI.CompMat,BB_IPI.GivenDisplacements);%Least-squares
             formula
    case 'constrained'
        %forcing C1 continuity
        kvec = (BB_IPI.TractionMesh(3:end)-BB_IPI.TractionMesh(2:end-1))...
            ./(BB_IPI.TractionMesh(2:end-1)-BB_IPI.TractionMesh(1:end-2));
        restmatlin = diag(kvec+1)+diag(-kvec(2:end),-1)-diag(ones([1 BB_IPI.NumLoads-1]),1);
        restmatquad = diag(2*kvec+1)+diag(-kvec(2:end),-1);
        Aeq = [restmatlin restmatquad];
        beq = zeros([BB_IPI.NumLoads 1]);
        %forcing null initial slope
        Aeq = [1 zeros([1 2*BB_IPI.NumLoads-1]);Aeq];
        beq = [0;beq];
        %forcing null final slope
        tmp = [zeros([1 BB_IPI.NumLoads-1]) 1];
        Aeq = [Aeq;tmp tmp];
        beq = [beq;0];
        %adding force column
        Aeq = [Aeq zeros(size(beq))];
        %forcing force equilibrium
        Aeq = [Aeq;(BB_IPI.TractionMesh(3:end)-BB_IPI.TractionMesh(1:end-2))/2,...
           diff(BB_IPI.TractionMesh(1:end-1))/3+diff(BB_IPI.TractionMesh(2:end))/2 ,1];
        beq = [beq;0];
        %forcing moment equilibrium
        linearnegpart = diff(BB_IPI.TractionMesh(2:end)).*(2*BB_IPI.TractionMesh(2:end-1)...
```

```
            +BB_IPI.TractionMesh(3:end))/6;
        Aeq = [Aeq;diff(BB_IPI.TractionMesh(1:end-1)).*(BB_IPI.TractionMesh(1:end-2)...
            +2*BB_IPI.TractionMesh(2:end-1))/6+linearnegpart,...
            diff(BB_IPI.TractionMesh(1:end-1)).*(BB_IPI.TractionMesh(1:end-2)...
            +3*BB_IPI.TractionMesh(2:end-1))/12+linearnegpart,BB_IPI.ProblemData.L];
        beq = [beq;0];
        %calculation
        BB_IPI.CalculatedTractions = lsqconstrained(BB_IPI.CompMat...
            ,BB_IPI.GivenDisplacements,Aeq,beq);
    otherwise
        error(['Error in the least-squares algorithm selection.' ...
            'The available algorithms are pseudoinverse, QR, MP and lsqminnorm'])
end
BB_IPI.ProblemData.F = BB_IPI.CalculatedTractions(end);
BB_IPI.CalculatedTractions = [0; BB_IPI.CalculatedTractions(1:end-1); 0];
end
```

I.2.5 lsqconstrained

This solves a linear least squares problem with linear constraints..

```
function solution = lsqconstrained(G,b,Z,d,method) %()
%   LSQCONSTRAINED finds a solution for the linear least-squares problem
%   with linear equality constraints.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Default values
if ~exist('method') || isempty(method)
    method = 'SVD';
end

%% Sizes calculation
m = size(G,1);
n = size(G,2);
p = size(Z,1);
if p>=n
    error(['The linear equality contraints matrix (Z) of the linear'...
        ' least-squares computation does not have more columns than rows.'...
        ' The system is not overdetermined. Consider using more degrees'...
        ' of freedom in the calculation or removing constraints.'])
end

%% Method selection and calculation
switch method
    case 'SVD' %Singular Value Decomposition
        [U,V,X,C,S] = gsvd(G,Z);
        q = size(find(diag(C)==0),1);
        solution = X'\[V(:,1:p)'*d./diag(S(1:p,1:p));zeros([n-p 1])]...
            + X'\[zeros([p 1]);U(:,p+1:n)'*b./diag(C(p+1:n,p+1:n))];

    case 'QR' %QR decomposition
        [Q,R] = qr(Z');
        y = R(1:p,1:p)\d;
        AQ = G*Q;
        z = AQ(:,p+1:n)\(b-AQ(:,1:p)*y);
        solution = Q(:,1:p)*y+Q(:,p+1:n)*z;

    otherwise
        errror(['The specified algorithm for performing the constrained'...
            ' linear least-squares computation is not amongst the ones '...
            'available. The options are SVD and QR.'])

end
end
```

I.2.6 PositionOpt

In this function, the linear inverse parameter identification routine *InvParamId* is embedded in the nonlinear optimization function *fmincon* to find the optimum position of the calculated tractions with the considered tractions.

```
function BB_IPI = PositionOpt(BB_IPI)
%   POSITIONOPT finds the optimal position of the interpolation points of
%   the traction distribution where the traction on a beam are calculated
%   by an invese parameter procedure.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Mesh and limits
BB_IPI.GivenDisplacements = BB_IPI.GivenDisplacementsOrg(BB_IPI.DispLims(1)...
    <=BB_IPI.DispMeshOrg & BB_IPI.DispMeshOrg<=BB_IPI.DispLims(2));
BB_IPI.DispMesh = BB_IPI.DispMeshOrg(BB_IPI.DispLims(1)<=BB_IPI.DispMeshOrg...
    & BB_IPI.DispMeshOrg<=BB_IPI.DispLims(2));

%% Nonlinear optimization parameters
A = [zeros(1,BB_IPI.NumLoads+2); eye(BB_IPI.NumLoads+2)] - ...
        [eye(BB_IPI.NumLoads+2); zeros(1,BB_IPI.NumLoads+2)];
b = [-BB_IPI.LoadLims(1); zeros(BB_IPI.NumLoads+1,1)-1e-10; BB_IPI.LoadLims(2)];

%% Nonlinear optimization
switch BB_IPI.NonLinAlgorithm
    case 'none'
        BB_IPI.TractionMesh = linspace(BB_IPI.LoadLims(1),BB_IPI.LoadLims(2),BB_IPI.NumLoads+2);

    case 'fmincon'
        options = optimoptions('fmincon','Algorithm','interior-point','display','off',...
            'StepTolerance',1e-22,'OptimalityTolerance',1e-20,'MaxFunctionEvaluations',3e4);

        initialguess = linspace(BB_IPI.LoadLims(1),BB_IPI.LoadLims(2),BB_IPI.NumLoads+2);
        [BB_IPI.TractionMesh,BB_IPI.nonlin.fval,BB_IPI.nonlin.exitflag,BB_IPI.nonlin.output,...
            BB_IPI.nonlin.lambda,BB_IPI.nonlin.grad,BB_IPI.nonlin.hessian]...
            = fmincon(@(x) ObjFun(BB_IPI,x),initialguess,A,b,[],[],[],[],[],options);

    case 'ga'
        options = optimoptions('ga','display','iter','UseParallel',true,...
            'CrossoverFraction',0.01,'TolFun',1e-12,'TolCon',0);

        [BB_IPI.TractionMesh,BB_IPI.nonlin.fval,BB_IPI.nonlin.exitflag]...
            = ga(@(x) ObjFun(BB_IPI,x),BB_IPI.NumLoads+2,A,b,[],[],[],[],[],options);

    case 'multistart'
        trials = 20;
        x0mat = sort(rand(trials,BB_IPI.NumLoads+2)*(max(BB_IPI.DispMesh)...
            -min(BB_IPI.DispMesh)),2)+min(BB_IPI.DispMesh);
        tpoints = (CustomStartPointSet(x0mat));
        options = optimoptions('fmincon','Algorithm','interior-point','display','off');
        initialguess = linspace(BB_IPI.LoadLims(1),BB_IPI.LoadLims(2),BB_IPI.NumLoads+2);
        problem = createOptimProblem('fmincon','x0',initialguess,'objective',@(x) ObjFun(BB_IPI,x),...
            'options',options,'Aineq',A,'bineq',b);
        solver = MultiStart('Display','off','StartPointsToRun','all','UseParallel',true);
        [BB_IPI.TractionMesh,BB_IPI.nonlin.fval,BB_IPI.nonlin.exitflag,BB_IPI.nonlin.output...
            ,BB_IPI.nonlin.solutions] = run(solver,problem,tpoints);

```

```
    otherwise
        error(['The selected algorithm for the tractions position optimization (NonLinAlg) is not '...
            'on the list of available algorithms. Choose fmincon or ga']);
end

BB_IPI = InvParamId(BB_IPI);

end
function error = ObjFun(BB_IPI,x)%objective function
BB_IPI.TractionMesh = x;
[BB_IPI] = InvParamId(BB_IPI);
BB_IPI.CalculatedDisplacements = BB_IPI.CompMat*[BB_IPI.CalculatedTractions(2:end-1);
     BB_IPI.ProblemData.F];
error = norm(abs(BB_IPI.GivenDisplacements - BB_IPI.CalculatedDisplacements));
end
```

I.2.7 InTraFiCa

This scripts organizes the input data in a *struct* variable called *BB_IPI* to facilitate the later calculations and data transfer, and the options of the calculation are set depending on the input defined by the user. In addition, the default values of the options are specified as well as some input checks.

```
function BB_IPI = InTraFiCa(ProblemData,Displacements,varargin)%()
%   INTRAFICA is a program made to perform the inverse parameter
%   identification of the interfacial tractions from the DIC measurements
%   of a DCB specimen.
%
%   It must be called as INTRAFICA(PROBLEMDATA,DISPLACEMENTS,OPTIONS) where
%   the different inputs are:
%
%       - PORBLEMDATA: a struct variable containing information of the beam
%       with the fields
%           * w: width of the beam.
%           * h: height of the beam.
%           * EI: flexural ridigity.
%           * kAG: shear ridigity.
%           * L: point of application of the external force.
%
%       - DISPLACEMENTS: a numeric matrix of two columns. In the first
%       column, the position of the points where the calculated
%       displacementsis stored. And the second column contains the vertical
%       displacement values at this points.
%
%       - OPTIONS: set options of the calculation as pairs of values
%       INTRAFICA(PROBLEMDATA,DISPLACEMENTS,'NAME1',OPTIONS1,...).
%
%   The results are returned in a struc called BB_IPI with multiple fields
%   having information about the calculation.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Initialization and default values
BB_IPI.NumLoads = DefaultValuesList(varargin,'NumberLoads',7);
BB_IPI.LinAlgorithm = DefaultValuesList(varargin,'LinAlg','constrained');
BB_IPI.LoadLims = DefaultValuesList(varargin,'TracLoc',[0 ProblemData.L]);
BB_IPI.DispLims = DefaultValuesList(varargin,'DispLoc',[0 ProblemData.L]);
BB_IPI.NonLinAlgorithm = DefaultValuesList(varargin,'NonLinAlg','fmincon');
BB_IPI.initialguess = DefaultValuesList(varargin,'InitialGuess', []);
BB_IPI.ComplianceFun = DefaultValuesList(varargin,'CompFun',@CombinedCompliance);

%% Struct form organization
BB_IPI.ProblemData = ProblemData;
BB_IPI.DispMeshOrg = Displacements(:,1);
BB_IPI.GivenDisplacementsOrg = Displacements(:,2);

%% Initial parameters
BB_IPI.GivenDisplacements = BB_IPI.GivenDisplacementsOrg(BB_IPI.DispLims(1)...
    <=BB_IPI.DispMeshOrg & BB_IPI.DispMeshOrg<=BB_IPI.DispLims(2));
BB_IPI.DispMesh = BB_IPI.DispMeshOrg(BB_IPI.DispLims(1)<=BB_IPI.DispMeshOrg ...
    & BB_IPI.DispMeshOrg<=BB_IPI.DispLims(2));

%% Tractions identification
BB_IPI = PositionOpt(BB_IPI);
```

```
BB_IPI.CalculatedDisplacements = BB_IPI.CompMat*...
    [BB_IPI.CalculatedTractions(2:end-1); BB_IPI.ProblemData.F];

%% Postprocessing to obtained the traction field
BB_IPI.TractionMeshOrg = BB_IPI.TractionMesh;
BB_IPI.CalculatedTractionsOrg = [BB_IPI.CalculatedTractions(1:end/2);0];

x = linspace(BB_IPI.TractionMesh(1),BB_IPI.TractionMesh(end),1e4)';
gfun1 = @(x) x;
gfun2 = @(x) 1-x;
cond1 = x>=BB_IPI.TractionMesh(1:end-2) & x<=BB_IPI.TractionMesh(2:end-1);
cond2 = x>BB_IPI.TractionMesh(2:end-1) & x<=BB_IPI.TractionMesh(3:end);
matcoord1 = (x-BB_IPI.TractionMesh(1:end-2))./(BB_IPI.TractionMesh(2:end-1)-BB_IPI.TractionMesh(1:end-2));
matcoord2 = (x-BB_IPI.TractionMesh(2:end-1))./(BB_IPI.TractionMesh(3:end)-BB_IPI.TractionMesh(2:end-1));
matdatalin = (cond1.*gfun1(matcoord1)+cond2.*gfun2(matcoord2)).*BB_IPI.CalculatedTractionsOrg(2:end-1)';

BB_IPI.TractionMesh = x';

%% Traction shape quad
BB_IPI.CalculatedTractionsOrg = [0;BB_IPI.CalculatedTractions((end/2+1):end)]...
    + BB_IPI.CalculatedTractionsOrg;
gfun1 = @(x) x.^2;
gfun2 = @(x) 1-x;
matdataquad =
     (cond1.*gfun1(matcoord1)+cond2.*gfun2(matcoord2)).*BB_IPI.CalculatedTractions((end/2+1):end-1)';

BB_IPI.CalculatedTractions = sum(matdatalin,2) + sum(matdataquad,2);
end
function output = DefaultValuesList(input,name,value)
if any(ismember(name,input(1:2:end)))
    output = input{find(ismember(input(1:2:end),name))*2};
else
    output = value;
end
end
```

I.2.8 CLextracter

This function is used to automatically obtain the cohesive law from a traction distribution and a displacement distribution.

```
function [fullcl truepoints] = CLextracter(BB_IPI,disp,tol,numpoints)
%   CLEXTRACTER extracts a cohesive law from the values of a displacement
%   and stress at the delamination interface.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.

%% Variable simplification
stress = [BB_IPI.TractionMesh',-BB_IPI.CalculatedTractions/BB_IPI.ProblemData.w];
[a pos] = max(stress(:,2));

%% Default values
if ~exist('tol') || isempty(tol)
    tol = max(stress(:,2))/1e3;
end
if ~exist('numpoints') || isempty(numpoints)
    numpoints = 1e4;
end

%% Area with active cohesive law determination

% Search of the first negative stress in the direction of the clamp
tmp = inf;
i = 0;
while tmp>tol && pos-i~=1
    i=i+1;
    tmp = stress(pos-i,2);
end

% Search of the first negative stress in the load application direction
j=0;
tmp = inf;
while tmp>tol && pos~=length(stress(:,2))
    j=j+1;
    tmp = stress(pos+j,2);
end

limits = [stress(pos-i,1) stress(pos+j,1)];

% Take only the tractions where damage is occuring.
if limits(2) > max(disp(:,1))
   limits(2) = max(disp(:,1));
end
if limits(1) < min(disp(:,1))
   limits(1) = min(disp(:,1));
end
%% Interpolation: find traction-separation values
x = linspace(limits(1),limits(2),numpoints)';
fullcl = [interp1(disp(:,1),disp(:,2),x) ...
    interp1(stress(:,1),stress(:,2),x)];
xtrue = BB_IPI.TractionMeshOrg(limits(2)>=BB_IPI.TractionMeshOrg ...
    & BB_IPI.TractionMeshOrg>=limits(1))';
truepoints = [interp1(disp(:,1),disp(:,2),xtrue) ...
    -BB_IPI.CalculatedTractionsOrg(limits(2)>=BB_IPI.TractionMeshOrg...
```

```
    & BB_IPI.TractionMeshOrg>=limits(1))/BB_IPI.ProblemData.w];

%% Taking FPZ stresses only
sepmax = fullcl(fullcl(:,2) == max(fullcl(:,2)),1);
truepoints = truepoints(truepoints(:,1)>=sepmax,:);
fullcl = fullcl(fullcl(:,1)>=sepmax,:);

%% Correction of the displacement values
truepoints(:,1) = 2*(truepoints(:,1)-min(fullcl(:,1)));
fullcl(:,1) = 2*(fullcl(:,1)-min(fullcl(:,1)));

%% Removing negative stresses
truepoints = truepoints(truepoints(:,2)>=0,:);
stresscondition = fullcl(:,2)>=0;
fullcl = fullcl(stresscondition,:);

end
```

I.3 Monte Carlo Analyses

I.3.1 CLsampling

This script generates a random sample of the input data of the inverse parameter identification each time is called.

```
function [PData DData DDatainter ValuesTaken] = CLsampling(PerData)
%   CLSAMPLING generated random values of different parametes acroding to a
%   distribution specified in PerData. The output is used for a Monte Carlo
%   simulation.
%
%   Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%   Master in Design of Mechanical Systems, Aalborg University, Denmark.


% Loop over the elements
for i = 1:length(PerData.names)
    %% Generate a random number according to the distribution
    if strcmp(PerData.dist{i},'N')
        RandVal = PerData.values(1,i) + randn*PerData.values(2,i)*PerData.values(1,i);
    elseif strcmp(PerData.dist{i},'Ui')
        RandVal = randi(diff(PerData.values(:,i))+1)+PerData.values(1,i)-1;
    else
        error('Not valid probability distribution. The types are N, U and Ui.')
    end

    %% Create the values for the analysis with the random number
    switch PerData.names{i}
        case 'EI'
            PData.EI = RandVal;
            ValuesTaken(i) = PData.EI;

        case 'kAG'
            PData.kAG = RandVal;
            ValuesTaken(i) = PData.kAG;

        case 'L'
            PData.L = RandVal;
            ValuesTaken(i) = PData.L;

        case 'w'
            PData.w = RandVal;
            ValuesTaken(i) = PData.w;

        case 'PicNum'
            DData = PerData.disp{RandVal};
            DDatainter = PerData.dispinter{RandVal};
            ValuesTaken(i) = RandVal;

        otherwise
            error('Not valid input parameter for MC.')
    end

end

%% Store random values generated
ValuesTaken = ValuesTaken';
```

```
end
```

I.3.2 MonteCarlo Main

This script is the one which calls the different functions to perform the Monte Carlo analyses. It contains the loop which performs the repeated sampling by calling the function *CLsampling* to generate the random input data and it calls *InTraFiCa* after that to perform the inverse parameter identification of the cohesive corresponding to this set of random values.

```
%%%%% Monte Carlo algorithm to evaluate the precision of a cohesive law
%%%%% calculated by the routine InTraFiCa. The calculations performed by
%%%%% this script are computationally expensive. It is prepared to run in
%%%%% parallel, so a cluster with several cores is strongly recomended for
%%%%% performance.
%%%%%
%%%%% Pere Joan Jaume Camps, Felix Prieto Viejo and Jose Mato Sanz [2020].
%%%%% Master in Design of Mechanical Systems, Aalborg University, Denmark.

clear;clc;close all
%% Inverse parameter identification program
restoredefaultpath
% selection of the folder which contains the scripts of the model to use
LoadingModelFolder = 'Routines_lin';
addpath(LoadingModelFolder)

%% Model parameters definition
% Random variables definition
PerturbationData.names = {'EI','kAG','L','w','PicNum'};
PerturbationData.dist = {'N','N','N','N','Ui'};
  %N: [mean;sigma/mean], Ui: [lowerbound;upperbound]
EImean = 0.0278*0.00425^3/12*4.595e10;
kAGmean = 5/6*2.2e9*0.0278*0.00425;
PerturbationData.values = [EImean,kAGmean,0.64,0.0278,1;
                           0.044,0.1,5e-4,0.002,51];
% DIC displacement data
load DICdata/DisplacementData_Size15_Test_long2
PerturbationData.disp = DisplacementData;
PerturbationData.dispinter = DisplacementData;

%% Analysis
% vector of delta_j
fixedseparation = linspace(0,2e-3,1e4)';
colorlist = {'b','r','g','c','m','y'};
% parameter to change in between different calculations
paramchange = [2 3 5 7 8 10 12];

for i=1:length(paramchange)
    % Values initialization for the parallel calculation
    AllCL = [];
    AllParamVals = [];
    nonlininfo = [];
    CalcForce = [];
    condnum = [];
    Nloads = paramchange(i);
    itercont = 0;
    change1 = 1;
    change2 = 1;
    currentmean = zeros(length(fixedseparation),1);
    currentconflims = zeros(length(fixedseparation),2);
    changehist1 = 1;
    changehist2 = [1 1];
```

```

% handle function to the inverse parameter identification routine
IPI_routine = @(x,y) InTraFiCa(x,y,'NumberLoads',Nloads,'TracLoc',[0.416 0.61]);

%Loop until stopping cirteria
while (change1 > 0.01 | max(change2) > 0.01)

    itercont = itercont + 1;

    % Calculation
    parallelcalcs = 40;%number of calculations in between
    parfor j = 1:parallelcalcs
        %Generation of random values
        [ProblemData DispData, DispDatainter paramvals] = CLsampling(PerturbationData);
        %Inverse parameter identification of the tractions
        BB_IPI = IPI_routine(ProblemData,DispData);
        % CL calculation and interpolation to delta_j
        CLfull = CLextracter(BB_IPI,DispDatainter);
        CLpoints = interp1(CLfull(:,1),CLfull(:,2),fixedseparation);
        % Storing information
        AllCL = [AllCL CLpoints];
        AllParamVals = [AllParamVals paramvals];
        CalcForce = [CalcForce BB_IPI.ProblemData.F];
        if isfield(BB_IPI,'CalculatedTractionsOrg')
            BB_IPI.nonlinposition = BB_IPI.TractionMeshOrg;
        end
        nonlininfo= [nonlininfo BB_IPI.nonlin];
        condnum = [condnum cond(BB_IPI.CompMat)];

    end

    tmpAllCL = AllCL(:,1+end-parallelcalcs:end);
    tmpAllCL(isnan(tmpAllCL)) = 0;
    AllCL(:,1+end-parallelcalcs:end) = tmpAllCL;

    % Stopping criteria value calculation
      % Mean
    tmpmean = mean(AllCL,2);
    changemean = abs((tmpmean - currentmean)./tmpmean);
    currentmean = tmpmean;
    change1 = max(changemean);
    changehist1(itercont,:) = change1;

      % Confidence interval
    confidencelevel = 0.95;
    sortmat = sort(AllCL,2);
    maxdim = size(AllCL,2);
    tmpconflims = sortmat(:,round([(1-confidencelevel)/2*maxdim (1+confidencelevel)/2*maxdim]));
    changeconflims = abs((tmpconflims - currentconflims)./[mean(currentmean) mean(tmpconflims(:,2))]);
    currentconflims = tmpconflims;
    change2 = max(changeconflims);
    changehist2(itercont,:) = change2;


    %Stopping criteria plots
    figure(i)
    subplot(2,2,1)
    semilogy(fixedseparation,[changemean changeconflims])
    legend('mean','lower lim','upper lim')
    title(['Change each iteration for ' num2str(Nloads) ' loads'])
```

```
        axis([-1e-5 1.6e-4 0 1.2*max([change1 change2])])
        subplot(2,2,2)
        semilogy(parallelcalcs:parallelcalcs:itercont*parallelcalcs,changehist1)
        title(['Maximum change in mean and std for each iteration for ' num2str(Nloads) ' loads'])
        legend('mean')
        subplot(2,2,3)
        semilogy(parallelcalcs:parallelcalcs:itercont*parallelcalcs,changehist2)
        legend('lower','upper')
        title(['Maximum change in conflims each iteration for ' num2str(Nloads) ' loads'])
        subplot(2,2,4)
        plot(fixedseparation,currentmean,'b','LineWidth',1.5)
        hold on
        plot(fixedseparation,currentconflims,'b--')
        hold off
        legend('main','lower','upper')
        title('Actual cohesive law')
        drawnow
        clc
        [changehist1 changehist2]
    end
    %store results in a struct and save them to the current folder
    clearvars IPImodel_analysis
    IPImodel_analysis.CLall = AllCL;
    IPImodel_analysis.CLmean = currentmean;
    IPImodel_analysis.CLconflims = currentconflims;
    IPImodel_analysis.Conflevel = confidencelevel;
    IPImodel_analysis.Parameters.values = AllParamVals;
    IPImodel_analysis.Parameters.names = PerturbationData.names;
    IPImodel_analysis.handlefun = IPI_routine;
    IPImodel_analysis.changehist = [changehist1 changehist2];
    IPImodel_analysis.separation = fixedseparation;
    IPImodel_analysis.nonlin = nonlininfo;
    IPImodel_analysis.force = CalcForce;
    IPImodel_analysis.description = ['Descrition of the calculation'];

    save(['IPImodel_analysis_' num2str(Nloads) 'dof_name'],'IPImodel_analysis')

    %Plot of the different CLs calculated
    figure(length(paramchange)+1)
    meanref = plot(fixedseparation,currentmean,'LineWidth',1.5);
    hold on
    a = plot(fixedseparation,currentconflims,['--']);
    a(1).Color = meanref.Color;
    a(2).Color = meanref.Color;
    hold on

    if ~exist('legendlist')
        legendlist.subsets = meanref;
        legendlist.names = {[num2str(Nloads) ' loads']};
    else
        legendlist.subsets = [legendlist.subsets meanref];
        legendlist.names = [legendlist.names [num2str(Nloads) ' loads']];
    end
end
```